Официальное руководство

по подготовке к сертификационным экзаменам

Cisco CCENT/ CCNA ICND1 100-101

Академическое издание

www.williamspublishing.com
www.ciscopress.ru
ciscopress.com

УЭНДЕЛЛ ОДОМ, CCIE® №1624

Cisco CCENT/ CCNA ICND1 100-101 **Official** Cert Guide

Academic Edition

WENDELL ODOM, CCIE NO. 1624

Cisco Press
800 East 96th Street
Indianapolis, IN 46240

Официальное руководство
по подготовке к сертификационным экзаменам

Cisco CCENT/ CCNA ICND1 100-101

Академическое издание

УЭНДЕЛЛ ОДОМ, CCIE® №1624

Москва • Санкт-Петербург • Киев
2015

Издательский дом "Вильямс"
Зав. редакцией *С.Н. Тригуб*
Перевод с английского и редакция *В.А. Коваленко*
По общим вопросам обращайтесь в Издательский дом "Вильямс" по адресу:
info@williamspublishing.com, http://www.williamspublishing.com

Одом, Уэнделл.
Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101, акад. изд. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2015. — 912 с. : ил. — Парал. тит. англ.

Научно-популярное издание
Уэнделл Одом
Официальное руководство Cisco по подготовке к сертификационнымэкзаменам CCENT/CCNA ICND1 100-101 Академическое издание

Литературный редактор *И.А. Попова*
Верстка *О.В. Мишутина*
Художественный редактор *В.Г. Павлютин*
Корректор *Л.А. Гордиенко*

Подписано в печать 19.02.2015. Формат 70x100/16
Гарнитура Times.
Усл. печ. л. 72,74. Уч.-изд. л. 54,6
Тираж 500 экз. Заказ № 866

Отпечатано способом ролевой струйной печати
в АО «Первая Образцовая типография»
Филиал «Чеховский Печатный Двор»
142300, Московская область, г. Чехов, ул. Полиграфистов, д.1

ООО "И. Д. Вильямс", 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1906-9 (рус.)
ISBN 978-1-58714485-1 (англ.)

Оглавление

Содержание

Об авторе

Уэнделл Одом, сертифицированный эксперт компании Cisco CCIE (Cisco Certified Internetwork Expert — сертифицированный эксперт по сетям компании Cisco) № 1624, работает в сфере сетевых технологий с 1981 года. Уэнделл работал сетевым инженером, консультантом, системным инженером, инструктором и принимал участие в разработке курсов по сетям, а ныне занимается проектированием и разработкой средств сертификации. Он является автором всех предыдущих редакций серии книг *CCNA Official Certification Guide* издательства Cisco Press для подготовки к экзаменам CCNA, книг по технологиям Cisco QOS и многих других, а также одним из авторов книги *CCIE Routing and Switch*. Уэнделл консультировал также компанию Pearson при подготовке ее новой версии эмулятора CCNA 640-802 Network Simulator. Он также поддерживает инструментальные средства обучения, ссылки на свои блоги и другие ресурсы на сайте `www.certskills.com`.

Посвящения

Памяти Уильяма Е. Йорка (William E. York), отца матери, или Пав Пава, носившего синий джинсовый комбинезон, возившегося с водопроводом, рыбачившего на озере Джулиет (Juliet Lake) на червяка. Скучаю по его громкому смеху.

Благодарности

Эта книга и сопутствующая ей книга по ICND2 представляют седьмое издание в серии книг издательства Cisco Pres, призванных помочь в сдаче экзамена на сертификат CCENT и CCNA Routing and Switching. С учетом столь длинной истории (первое издание вышло в 1998 году) над этими книгами поработало множество людей, которые осуществляли разработку, техническое и литературное редактирование, корректуру, индексацию, управление рабочим процессом, разрабатывали внутренний дизайн, дизайн обложки, занимались маркетингом и выполняли все те действия, без которых не выпустить книги. Спасибо вам всем за вашу роль в выпуске книги.

Большинство участников предыдущего издания, включая редактора проекта Дрю Каппа (Drew Cupp), работали и над нынешним изданием. Несмотря на мои частые коррективы содержимого и заголовков, Дрю удалось сохранить ясность всех деталей, упорядочивая их на ходу, т.е. он делал свою привычную работу: обеспечение простоты и единообразия текста и материалов по всей книге. Спасибо, Дрю, за то, что провел меня через этот путь.

Что касается технического редактирования, то Элан Бир сделал свою работу, как обычно, безупречно. Он находил небольшие ошибки в перекрестных ссылках на отдельные страницы, способные ввести читателя в заблуждение и затруднить понимание некоторых фраз. И так по всем техническим вопросам. Фантастическая работа! Спасибо, Элан.

Брет Бартоу (Brett Bartow) снова был исполнительным редактором книги, как и почти с самого начала выхода этих изданий. Если говорить о роли Брета за эти го-

ды, то лучше всего сказать "товарищ по команде". Брет мог бы работать в Pearson Education, но он всегда работает со мной, не упуская бизнес-вопросов и находя наилучшие способы отношения между издателем и автором. Спасибо за еще одну прекрасную работу над этой книгой, Брет!

Документы Word перебрасываются туда-сюда, пока не получается готовый, красивый текст. Только благодаря Сандре Шрёдер (Sandra Schroeder), Тоне Симпсон (Tonya Simpson) и всей рабочей группе стало возможно волшебство создания из этих документов Word готовой книги. Они сделали все: от исправления моей грамматики, подбора слов и оборотов речи до последующей сборки и компоновки проекта. Спасибо, что собрали все это вместе и красиво оформили. Тоня, на удивление удачно, жонглировала сотнями элементов двух книг по CCNA, одновременно управляя несколькими процессами. Спасибо за это! И отдельное спасибо за внимание к подробностям.

Процесс подготовки рисунков для этих книг проходил немного не так, как для других книг. Совместно мы приложили массу усилий по модернизации рисунков обеих книг, как по дизайну, так и по содержимому, а также предоставили цветные версии для электронных книг. Особая благодарность Лоре Роббинс (Laura Robbins) за работу над цветом и стандартами дизайна в этом процессе. Кроме того, благодарю Майка Танамаши (Mike Tanamachi) за рисунки и их переделку после каждых моих изменений.

Благодарю Криса Бернса (Chris Burns) из CertSkills за работу над задачами, используемыми как в приложениях, так и в книге, а также за проверку некоторых из глав.

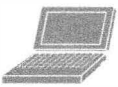
Особая благодарность читателям, которые высказывали свои предложения, находили ошибки, а особенно тем из вас, кто писал сообщения в учебную сеть Cisco (Cisco Learning Network — CLN). Без сомнения, те комментарии, которые я получал лично и читал в сети CLN, сделали это издание лучше.

Благодарю свою жену Крис. Жесткий график написания книги серьезно повлиял на то, чего я хотел, но не всегда мог. Благодарю мою дочь Ханну за все исследования и работу, мешающие иногда школьным занятиям. Благодарю Иисуса Христа за возможность писать.

Условные обозначения сетевых устройств


Принтер


ПК


Портативный ПК


Сервер

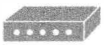

Телефон


IP-телефон


Маршрутизатор


Коммутатор

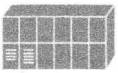

Коммутатор Frame Relay

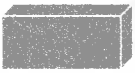

Кабельный модем


Точка доступа


ASA


DSLAM


Коммутатор WAN


CSU/DSU


Концентратор


Брандмауэр PIX


Мост


Коммутатор третьего уровня


Сетевая среда

Соединение Ethernet


Последовательный канал

Виртуальный канал


WAN Ethernet


Беспроводное соединение

Введение

Об экзаменах

Прежде всего, эта книга задумана как учебник для курса изучения сетей в колледже. В то же время, если желание сделать карьеру в области телекоммуникаций появилось позже, эта книга окажет существенную помощь в этом начинании, облегчив сдачу экзамена на сертификат Cisco.

Если вы дочитали эту книгу до введения, то наверняка решили получить сертификат специалиста компании Cisco. Чтобы добиться успеха на поприще технического специалиста в сетевой индустрии, современный сетевой инженер должен быть знаком с оборудованием компании Cisco. Компания имеет невероятно высокую долю на рынке оборудования для маршрутизации и коммутации — в общем, более 80% в некоторых регионах. Во многих странах и на мировом рынке синонимом слова "сеть" является название компании Cisco. Если читатель хочет, чтобы к нему относились как к серьезному сетевому специалисту, то имеет смысл получить сертификацию компании Cisco.

Экзамены, позволяющие получить сертификаты CCENT и CCNA

Компания Cisco объявила об изменениях в сертификации CCENT и CCNA Routing and Switching, а также связанных с ними экзаменах 100-101 ICND1, 200-101 ICND2 и 200-120 CCNA в начале 2013 года. Для тех, кто знает, как сдавали прежние экзамены Cisco ICND1, ICND2 и CCNA: структура осталась той же. Для новичков в сертификации Cisco данное введение начинается с обсуждения основ.

Почти все новички в сертификации Cisco начинают с сертификата CCENT или CCNA Routing and Switching. Сертификат CCENT требует примерно половины знаний и квалификации, необходимых для сертификата CCNA Routing and Switching. Таким образом, сертификат CCENT — это более простой первый этап.

Сертификация CCENT требует только одного этапа: сдачи экзамена ICND1. Достаточно просто.

Для получения сертификата CCNA Routing and Switching есть две возможности, как показано на рис. 1.1: сдать экзамены ICND1 и ICND2 либо сдать только один экзамен CCNA. (Обратите внимание: для сдачи экзамена ICND2 нет никакой отдельной сертификации.)

Как можно заметить, хотя сертификат CCENT можно получить, сдав экзамен ICND1, вовсе необязательно иметь сертификат CCENT, чтобы получить сертификат CCNA Routing and Switching. Можно сдать экзамен CCNA и пропустить сертификацию CCENT.

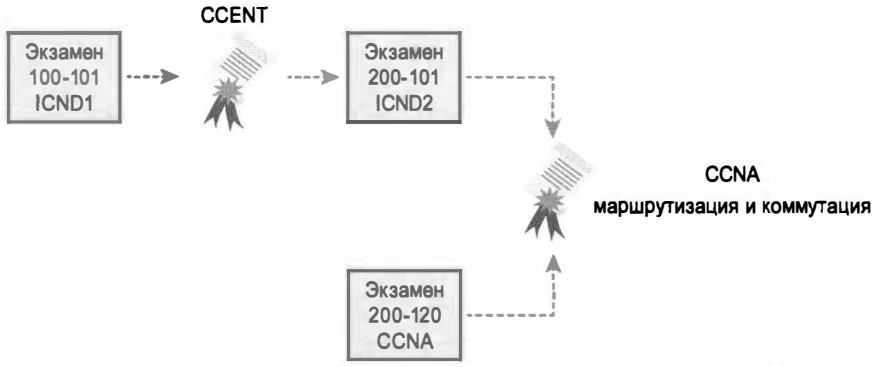


Рис. 1.1. Сертификация и экзамены начального уровня компании Cisco

Что касается самих тем экзаменов ICND1 и ICND2, то они разные, лишь с небольшим количеством совпадений. Например, экзамен ICND1 рассматривает основы открытого протокола поиска первого кратчайшего маршрута (Open Shortest Path First — OSPF). На экзамене ICND2 протокол OSPF рассматривается более подробно, но обсуждение этих дополнительных подробностей полагается на основы, полученные на экзамене ICND1. Многие из тем экзамена ICND2 полагаются на темы экзамена ICND1, вызывая некоторое перекрытие материала.

Экзамен CCNA включает все темы экзаменов ICND1 и ICND2 — ни больше ни меньше.

Типы экзаменационных вопросов

Экзамены ICND1, ICND2 и CCNA имеют одинаковый формат. В центре сертификации претендент находится в тихой комнате наедине с компьютером. Прежде чем начнется экзамен, у каждого будет шанс решить несколько других задач, например, можно решить примеры контрольных вопросов только для того, чтобы привыкнуть к компьютеру и механизму проверки. У любого, обладающего квалификацией пользователя персонального компьютера, не должно быть никаких проблем с экзаменационной системой.

После начала экзамена вопросы на экране появляются один за другим. Обычно они относятся к одной из следующих категорий:

- многовариантный выбор (Multiple Choice — MC) одного ответа;
- многовариантный выбор нескольких ответов;
- тестлет (testlet);
- вопросы с перетаскиванием правильных ответов (Drag-and-drop — DND);
- лабораторная работа на эмуляторах оборудования (Simulated lab — Sim);
- симлет (simlet).

Первые три типа вопросов в списке — фактически выбор правильного ответа. Многовариантный формат требует указать или щелкнуть на кружке около правильного ответа (ответов). Экзаменационное программное обеспечение компании Cisco

традиционно указывает количество правильных ответов и не позволит выбрать слишком много ответов. Тестлеты — это вопросы с одним общим сценарием и многовариантными вопросами в общем сценарии.

Вопросы с перетаскиванием ответов (DND) требуют перемещения мышью элементов GUI. Следует нажать левую кнопку мыши и, не отпуская ее, переместить пиктограмму или кнопку на экране в другое место, а затем отпустить кнопку мыши, чтобы расположить объект где-либо в другом месте (обычно в списке). Иногда, например, чтобы дать правильный ответ, придется расположить до пяти объектов в правильном порядке!

В последних двух случаях используется эмулятор сети. Следует отметить, что в действительности эти два типа вопросов позволяют компании Cisco оценивать два совсем разных навыка. В первом типе заданий описывается ошибка и стоит задача настроить один или несколько маршрутизаторов и коммутаторов, чтобы устранить проблему. В экзамене такое задание оценивается по той конфигурации, которая была сделана, или по изменениям, внесенным в существующую конфигурацию.

Симлеты — одни из наиболее сложных экзаменационных вопросов. В симлетах также используются эмуляторы сети, но вместо ответа на вопрос или изменения конфигурации в них нужно дать один или несколько многовариантных ответов. В таких вопросах нужно использовать эмулятор для проверки текущего поведения сети, интерпретации информации, выводимой командами группы `show`, которые экзаменуемый сможет вспомнить, чтобы ответить на вопрос. Если вопросы с эмуляцией сети требуют от специалиста умения диагностировать неисправности на основе конфигурации, то симлеты требуют умения проанализировать как исправную сеть, так и неисправную, связать команды группы `show` со знанием теории сети и конфигурационных команд.

Используя экзаменационный учебник Cisco (Cisco Exam Tutorial), можно просмотреть и даже опробовать эти типы команд. Сертификационный учебник Cisco (Cisco Certification Exam Tutorial) можно найти на сайте `www.cisco.com`, если ввести "exam tutorial".

Как проводится экзамен CCNA

Помню, когда я еще учился в школе, после того как учитель объявлял о том, что скоро у нас будет тест или контрольная, кто-нибудь всегда спрашивал: "А что это будет за тест?" Даже в колледже студенты всегда хотят иметь больше информации о том, что именно будет на экзамене. Информация в таком случае добывается главным образом с вполне практической целью — знать, что нужно учить больше, что меньше, а что можно совсем не учить.

Компания Cisco вполне открыто предоставляет темы каждого из экзаменов. Она хочет, чтобы были известны и темы экзаменов, а также какие именно знания и навыки потребуются для каждой темы при сдаче сертификационных тестов. Для этого компания Cisco публикует список, содержащий все темы.

Многие из экзаменационных вопросов Cisco включают темы по сети и описания. Описания свидетельствуют, до какой степени тема должна быть понятна и какие навыки необходимы. Задание подразумевает также наличие определенных навыков. Например, одно задание может начинаться так: "Опишите..." или так:

"Опишите, настройте и устраните неисправности...". Из постановки задачи в других заданиях можно четко понять, что необходимо полное понимание темы. Публикуя темы и необходимый уровень навыков для них, компания Cisco помогает специалистам готовиться к экзамену.

Несмотря на то что списки тем для экзаменов весьма полезны, не забывайте, что компания Cisco при публикации списка указывает, что он является *рекомендованным* набором тем для изучения. Компания Cisco стремится в экзаменационных вопросах не выходить за рамки таких тем, и специалисты, занимающиеся разработкой тестов, постоянно анализируют вопросы и обновляют их, чтобы они соответствовали заявленному списку.

Темы экзамена ICND1

Темы экзамена ICND1 перечислены в табл. 1.1–1.7, а в табл. 1.8–1.12 приведены темы экзамена ICND2. В этих таблицах отмечены главы, в которых затрагиваются данные экзаменационные темы.

Таблицы соответствуют организации тем Cisco и сгруппированы по темам и разделам. Разделы представляют более подробное описание специфических терминов и концепций тем экзаменационных задач. Основные темы в таблицах выделены полужирным шрифтом, а разделы набраны обычным.

Таблица I.1. Темы экзамена ICND1. Работа сетей передачи данных IP

Глава	Работа сетей передачи данных IP
1–4, 6, 15	**Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы**
1–4, 6, 15	**Выбор компонентов сети, удовлетворяющих заданной спецификации**
5	**Наиболее распространенные приложения и их воздействие на сеть**
1	**Описание предназначения и основных принципов протоколов в моделях OSI и TCP/IP**
2–5, 6, 9, 16, 24, 25	**Передача данных между двумя хостами по сети**
2, 6, 15	**Выбор подходящей среды, кабелей, портов и разъемов для подключения сетевых устройств Cisco к другим сетевым устройствам и хостам в сети LAN**

Таблица I.2. Темы экзамена ICND1. Технологии коммутации сетей LAN

Глава	Технологии коммутации сетей LAN
2, 6	**Технологии и методы управления доступом к передающей среде для сети Ethernet**
6, 9	**Базовые концепции коммутации и работа коммутаторов Cisco**
6	Домены коллизий
6, 9	Широковещательные домены
6	Типы коммутации
6, 9	Таблица CAM
7	**Настройка и проверка начальной конфигурации коммутатора, включая управление удаленным доступом**
7	Команды операционной системы Cisco IOS для базовой настройки коммутатора

Окончание табл. I.2

Глава	Технологии коммутации сетей LAN
7, 18, 28	**Проверка состояния сети и работоспособности коммутатора с помощью базовых сетевых утилит** `ping`, `telnet` **и** `ssh`
9	**Создание логических сегментов сети VLAN и необходимость маршрутизации между ними**
9	**Принцип сегментации сети и базовые концепции управления трафиком**
9	**Настройка и проверка сети VLAN**
9, 10	**Настройка и проверка магистрального соединения на коммутаторах Cisco**
9, 10	Протокол DTP
10	Автопереговоры

Таблица I.3. Темы экзамена ICND1. IP-адресация (IPv4/IPv6)

Глава	IP-адресация (IPv4/IPv6)
11	**Работа и необходимость использования частных и открытых IP-адресов при IPv4-адресации**
25, 26	**Выбор подходящей схемы IPv6-адресации, удовлетворяющей требованиям адресации в среде LAN/WAN**
11, 19, 20, 21	**Выбор подходящей схемы IPv4-адресации (использующей VLSM и суммирование), удовлетворяющей требования адресации в среде LAN/WAN**
27, 28, 29	**Техиологические требования для запуска протокола IPv6 совместно с протоколом IPv4 как двойного стека**
25–28	**Описание IPv6-адреса**
25, 26	Глобальный одноадресатный
27	Многоадресатный
27	Локальный адрес канала связи
26	Уникальный локальный адрес
27	Адрес в формате eui 64
28	Автоматическая настройка

Таблица I.4. Темы экзамена ICND1. Технологии маршрутизации IP

Глава	Технологии маршрутизации IP
16	**Базовые концепции маршрутизации**
16	CEF
16	Передача пакета
16	Процесс поиска маршрутизатора
15–18, 27	**Настройка и проверка применения CLI для установки базовой конфигурации маршрутизатора**
16–18, 27	Команды Cisco IOS для базовой настройки маршрутизатора
16, 27	**Настройка и проверка состояния интерфейса Ethernet**
16–18, 27–29	**Проверка конфигурации маршрутизатора и сетевого подключения**

Окончание табл. I.4

Глава	Технологии маршрутизации IP
16–18, 27, 29	Команды Cisco IOS для просмотра базовой информации маршрутизатора и сетевого подключения
16, 29	**Настройка и проверка конфигурации маршрутизации для статического или стандартного маршрута согласно заданным требованиям маршрутизации**
4, 16, 17, 25, 29	**Различия методов маршрутизации и протоколов маршрутизации**
4, 17, 29	Статика или динамика
17	Состояние канала или вектор расстояния
16, 25	Ближайшая точка перехода
16, 25	Таблица IP-маршрутизации
17, 29	Пассивные интерфейсы
17, 29	**Настройка и проверка OSPF (единая область)**
17, 29	Преимущество единой области
17	Настройка OSPF v2
29	Настройка OSPF v3
17, 29	Идентификатор маршрутизатора
17, 29	Пассивный интерфейс
16	**Настройка и проверка маршрутизации между VLAN (router on a stick)**
16	Субинтерфейсы
16	Восходящая маршрутизация
16	Инкапсуляция
8, 16	**Настройка интерфейсов SVI**

Таблица I.5. Темы экзамена ICND1. Службы IP

Глава	Службы IP
18, 28	**Настройка и проверка DHCP (маршрутизатор IOS)**
18, 28	Настройка интерфейса маршрутизатора для использования DHCP
18	Параметры DHCP
18	Исключенные адреса
18	Период резервирования
22, 23	**Типы, средства и приложения ACL**
22	Стандартные
23	Порядковые номера
23	Редактирование
23	Расширенные
23	Именованные
22, 23	Нумерованные
22	Средства регистрации
22, 23	**Настройка и проверка ACL в сетевой среде**
23	Именованные

Окончание табл. 1.5

Глава	Службы IP
22, 23	Нумерованные
22	Средства регистрации
24	**Базовые операции NAT**
24	Цель
24	Пул
24	Статический
24	1 к 1
24	Перегрузка
24	Исходная адресация
24	Односторонний NAT
24	**Настройка и проверка NAT для заданных требований сети**
23	**Настройка и проверка NTP на клиенте**

Таблица 1.6. Темы экзамена ICND1. Защита сетевых устройств

Глава	Защита сетевых устройств
8, 15, 23	**Настройка и проверка средств защиты сетевых устройств**
8, 15	Защита устройства паролем
8, 15	Привилегированный режим или защита
23	Транспорт
23	Отключение telnet
8	SSH
8	VTY
23	Физическая защита
8	Служебный пароль
8	Описание основных методов аутентификации
8, 10	**Настройка и проверка средств защиты порта коммутатора**
8	Автоматическое обнаружение MAC-адресов
8	Ограничение MAC-адресов
8, 10	Статические и динамические
8, 10	Реакция при нарушении защиты
8, 10	Отключение из-за ошибки
8, 10	Отключение
8, 10	Ограничение
8	Отключение неиспользуемых портов
8	Восстановление после ошибки
8	Присвоение неиспользуемых портов неиспользуемым VLAN
23	Установка собственной сети VLAN, отличной от VLAN 1
22, 23	**Настройка и проверка списков ACL для фильтрации сетевого трафика**
23	**Настройка и проверка списков ACL для ограничения обращений по протоколам telnet и SSH к маршрутизатору**

Таблица I.7. Темы экзамена ICND1. Поиск и устранение неисправностей

Глава	Поиск и устранение неисправностей
12–15, 18–21, 25–28	**Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресацни**
9, 10	**Поиск неисправностей и решение проблем сетей VLAN**
9, 10	Идентификация настроенных сетей VLAN
9, 10	Исправление принадлежности порта
9, 10	Настройка IP-адреса
9, 10	**Поиск неисправностей и решение проблем магистрального соединения на коммутаторах Cisco**
9, 10	Исправление состояния магистрального канала
9, 10	Исправление конфигурации инкапсуляции
9, 10	Исправление разрешенных VLAN
22, 23	**Поиск неисправностей и решение проблем списков ACL**
22, 23	Статистика
22, 23	Разрешенные сети
22, 23	Направление
22, 23	Интерфейс
10	**Поиск неисправностей и решение проблем уровня 1**
10	Фреймирование
10	CRC
10	Карлики
10	Гиганты
10	Отброшенные пакеты
10	Запоздалые коллизии
10	Ошибки ввода и вывода

Темы экзамена ICND2

Темы экзамена ICND2 приведены в табл. 1.8–1.12. Таблицы содержат ссылки на главы книги по ICND2, в которых затрагиваются темы экзамена. Обратите внимание, что в каждой таблице приведена основная тема экзамена. Информация по каждой теме разделена на несколько подуровней иерархии. В таблицах подуровни выделены отступами.

Таблица I.8. Темы экзамена ICND2. Технологии коммутации сетей LAN

Глава	Технологии коммутации сетей LAN
1	**Идентификация дополнительных технологий коммутации**
1	RSTP
1	PVSTP
1	EtherChannels
1, 2	**Настройка и проверка работы PVSTP**
1, 2	Описание выбора корневого моста
2	Режим связующего дерева

Таблица I.9. Темы экзамена ICND2. Технологии маршрутизации IP

Глава	Технологии маршрутизации IP
20	**Процесс загрузки операционной системы Cisco IOS маршрутизатора**
20	POST
20	Процесс загрузки маршрутизатора
12	**Настройка и проверка состояния последовательного интерфейса**
20, 21	**Управление файлами Cisco IOS**
20	Параметры загрузки
20	Образ (образы) Cisco IOS
21	Лицензии
21	Просмотр лицензии
21	Смена лицензии
8–11, 16–18	**Различия методов и протоколов маршрутизации**
8	Административное расстояние
9	Разделение диапазона
8, 9, 17, 18	Метрика
8, 9, 17, 18	Следующий транзитный узел
8, 17	**Настройка и проверка протокола OSPF (одиночная область)**
8, 11, 17	Соседские отношения
8, 11, 17	Состояние OSPF
8, 17	Несколько областей
8	Настройка OSPF v2
17	Настройка OSPF v3
8, 17	Идентификатор маршрутизатора
8, 17	Типы сообщений LSA
9, 10, 18	**Настройка и проверка EIGRP (одиночная область)**
9, 10, 18	Приемлемое расстояние / Возможные преемники / Административное расстояние
9, 18	Условие применимости
9, 18	Композиция метрик
9, 10, 18	Идентификатор маршрутизатора
9, 10	Автоматический отчет
9, 10, 18	Выбор пути
9, 10, 18	Баланс нагрузки
9, 10, 18	Равномерный
9, 10, 18	Неравномерный
9, 10, 18	Пассивный интерфейс

Таблица I.10. Темы экзамена ICND2. Службы IP

Глава	Службы IP
6	**Выявление технологии высокой доступности (FHRP)**
6	VRRP
6	HSRP
6	GLBP
19	**Настройка и проверка системного журнала**
19	Использование вывода системного журнала
19	**Описание протокола SNMP v2 и v3**

Таблица I.11. Темы экзамена ICND2. Поиск и устранение неисправностей

Глава	Поиск и устранение неисправностей
3–5, 16	**Поиск и устранение наиболее распространенных проблем сети**
19	**Использование данных сетевого потока**
2	**Поиск и устранение неисправностей в работе Resolve Spanning Tree**
2	Корневой коммутатор
2	Приоритет
2	Правильный режим
2	Состояние порта
4, 5, 16	**Поиск и устранение проблем маршрутизации**
4, 5, 16	Разрешение маршрутизации
4, 5, 16	Правильность таблицы маршрутизации
4, 5, 16	Выбор правильного пути
11, 17	**Поиск и устранение проблем OSPF**
11, 17	Соседские отношения
11, 17	Таймеры Hello и Dead
11, 17	Область OSPF
11, 17	Максимальный блок передачи данных интерфейса
11, 17	Типы сетей
11, 17	Состояние соседей
11, 17	База данных топологии OSPF
11, 18	**Поиск и устранение проблем EIGRP**
11, 18	Соседские отношения
11, 18	Номер AS
11, 18	Балансировка нагрузки
11, 18	Разделенный диапазон
3, 5	**Поиск и устранение проблем маршрутизации interVLAN**
5	Подключение
5	Инкапсуляция
5	Подсеть

Окончание табл. I.11

Глава	Поиск и устранение неисправностей
3, 5	Собственная сеть VLAN
3, 5	Состояние режима порта магистрального канала
12, 14	**Поиск и устранение проблем реализации WAN**
12	Последовательные интерфейсы
12	PPP
14	Frame Relay
19	**Контроль статистики NetFlow**
2	**Поиск и устранение проблем EtherChannel**

Таблица I.12. Темы экзамена ICND2. Технологии WAN

Глава	Технологии WAN
15, 13, 7	**Различные технологии WAN**
15	Metro Ethernet
15	VSAT
15	Сотовый 3G / 4G
15	MPLS
12, 15	T1 / E1
15	ISDN
15	DSL
13	Frame Relay
15	Кабель
7	VPN
12	**Настройка и проверка простого последовательного соединения WAN**
12	**Настройка и проверка соединения PPP между маршрутизаторами Cisco**
14	**Настройка и проверка Frame Relay на маршрутизаторах Cisco**
15	**Реализация и устранение проблем PPPoE**

Темы экзамена 200-120 CCNA

Экзамен 200-120 CCNA фактически покрывает весь материал экзаменов ICND1 и ICND2, по крайней мере, исходя из опубликованных экзаменационных тем. На момент написания книги экзамен CCNA включал все темы из табл. I.1–I.12. Короче говоря, CCNA = ICND1 + ICND2.

ВНИМАНИЕ!

Поскольку экзаменационные темы со временем могут измениться, имеет смысл перепроверить их на веб-сайте Cisco по адресу `www.cisco.com/go/ccent` или `www.cisco.com/go/ccna`. Если компания Cisco добавит впоследствии новые темы экзаменов, то в приложении Б, "Обновление экзамена ICND1", описано, как перейти на сайт `www.ciscopress.com` и загрузить дополнительную информацию о вновь добавленных темах.

О книге

В книге содержатся знания и навыки, необходимые для сдачи экзамена 100-101 ICND1. Ее содержимое составляет первую половину материала для экзамена CCNA ICND2, а вторая содержится во втором томе академического издания.

Особенности обеих книг одинаковы, поэтому, читая второй том после первого, нет необходимости читать совпадающее "Введение" повторно. Кроме того, если планируется использовать книги для подготовки к сдаче именно экзамена 200-120 CCNA, а не двух экзаменов последовательно, то имеет смысл ознакомиться с планом подготовки к экзамену, приведенным в конце данного раздела.

Особенности книги

Самая важная и вполне очевидная цель этой книги — помочь читателю получить знания и сдать экзамены ICND1 и CCNA. Изначально цель книги была несколько другой, поэтому название книги немного вводит в заблуждение. Тем не менее методы изложения материала, используемые в данной книге, несомненно, существенно помогут в сдаче экзаменов, а также помогут читателю стать высококвалифицированным специалистом в области информационных технологий и сетей.

В этой книге используется несколько средств, призванных помочь читателю обнаружить свои слабые места и темы, по которым следует улучшить свои знания и навыки, запомнить концептуальные моменты и дополнительные детали, а также разобраться в соответствующих технологиях досконально. Задача этой книги состоит не в том, чтобы помочь читателю сдать экзамен за счет зубрежки и хорошей памяти, а в том, чтобы обеспечить изучение и понимание ключевых технологий современных телекоммуникаций. Сертификат CCNA Routing and Switching является основой множества профессиональных сертификаций компании Cisco, поэтому книга ориентирована прежде всего на четкое понимание наиболее популярных стандартных технологий и протоколов. Книга поможет успешно сдать сертификационный экзамен CCNA, а также:

- поможет понять, какие темы экзамена следует изучить дополнительно;
- содержит информацию и подробные объяснения, которые помогут заполнить пробелы в знаниях;
- содержит упражнения, которые помогут запомнить материал и дедуктивным методом найти правильные ответы на экзаменационные вопросы;
- кроме того, на веб-странице книги по адресу: `http://www.williamspublishing.com/Books/978-5-8459-1906-9.html`, можно загрузить образ DVD-диска, содержащий практические примеры и задания по рассматриваемым темам, а также дополнительное тестовое программное обеспечение для подготовки к экзамену.

Особенности глав

Чтобы помочь читателю распланировать свое время в процессе изучения данной книги, в самых важных ее главах есть определенные элементы (см. ниже), которые помогут упорядочить процесс изучения материала.

- **Введение и темы экзамена.** Каждая глава начинается с введения, основных тем главы и списка тем официального экзамена, затронутых в этой главе.

- **Основные темы.** В этой основной части главы описаны протоколы, концепции и конфигурации, рассматриваемые в текущей главе.
- **Обзор.** Каждая глава завершается разделом "Обзор", в котором приведен набор подлежащих выполнению учебных действий. Каждая глава содержит те действия, которые наиболее подходят для изучения ее тем и могут включать следующие разделы.
  - **Резюме.** Полный перечень основных тем главы. Убедитесь в полном понимании всех этих пунктов. В противном случае повторно прочитайте главу.
  - **Контрольные вопросы.** Позволяют самостоятельно оценить свой уровень знаний по темам данной главы.
  - **Ключевые темы.** Соответствующая пиктограмма размещена рядом с самыми важными моментами каждой главы, а в конце главы приведена таблица ключевых тем. Несмотря на то что практически любой материал каждой главы может быть в экзамене, ключевые темы нужно знать особенно хорошо.
  - **Заполните таблицы и списки по памяти.** Чтобы помочь читателю натренировать память для уверенного запоминания информации и фактов, наиболее важные списки и таблицы вынесены в отдельное приложение на вебстранице книги. В другом приложении те же таблицы заполнены только частично, остальные записи читатель должен заполнить самостоятельно.
  - **Ключевые термины.** Хотя на экзаменах не попадаются вопросы, в которых нужно просто дать определение какого-либо термина, в экзамене CCNA требуется знание терминологии компьютерных сетей. В этом разделе перечислены основные термины главы, для которых нужно дать развернутые описания и сравнить их со списком терминов, который приведен в конце книги.
  - **Таблицы команд**. В некоторых главах описано множество команд конфигурации интерфейса командной строки. В таких таблицах перечислены команды, описанные в главе, наравне с их примерами, которые можно использовать как для запоминания команд, так и для подготовки к сертификационным экзаменам, при сдаче которых самые важные команды нужно помнить на память.

Обзор части

Этот раздел призван помочь в подготовке к практическому применению всех концепций данной части книги. (Каждая часть содержит несколько взаимосвязанных глав.) Обзор части включает примеры контрольных вопросов, требующих применения концепций из нескольких глав данной части, позволяя выяснить, действительно ли поняты все темы или не совсем. Здесь также приведены упражнения на проверку памяти, позволяющие научиться в уме объединять концепции, конфигурации и способы проверки, чтобы независимо от формулировки экзаменационного вопроса или конкретной конфигурации можно было проанализировать ситуацию и ответить на вопрос.

Наряду со списком задач в обзоре части содержатся контрольные вопросы, позволяющие проследить прогресс в обучении. Ниже приведен список наиболее распространенных задач, встречающихся в разделах обзоров частей; обратите внимание, что обзоры не всех частей содержат задачи каждого типа:

- **Повторите вопросы из обзора главы.** Хотя вопросы уже были представлены в обзорах глав, повторный ответ на те же вопросы в обзоре части может быть полезен. Раздел обзора части предлагает не только повтор вопросов из обзора главы, но и использование экзаменационного приложения PCPT, поставляемого вместе с книгой для дополнительной практики в ответах на вопросы с многовариантным выбором на компьютере.
- **Ответы на вопросы.** Экзаменационное приложение PCPT предоставляет несколько баз данных с вопросами. Одна экзаменационная база данных содержит вопросы, специально написанные для обзоров частей. Чтобы помочь в приобретении навыков, необходимых для более сложных вопросов об анализе на экзаменах, в каждый из данных вопросов включено по нескольку концепций, иногда из нескольких глав.
- **Ключевые темы.** Да, снова! Это, действительно, самые важные темы в каждой главе.
- **Конфигурационные диаграммы связей.** Диаграммы связей — это графические организационные инструменты, которые очень многие находят полезными при обучении и в работе для уяснения взаимодействия различных концепций. Процесс создания диаграмм связей поможет мысленно построить взаимосвязи между концепциями и командами конфигурации, а также выработать понимание отдельных команд. Диаграмму связей можно создать на бумаге или при помощи любого графического программного обеспечения на компьютере. (Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения данной книги.)
- **Проверочные диаграммы связей.** Эти упражнения призваны помочь соотнести команды `show` маршрутизатора и коммутатора с сетевыми концепциями или командами конфигурации. Диаграммы связей можно создать на бумаге или с помощью любого подходящего программного обеспечения.
- **Повтор задач из обзора главы.** (Необязательно.) Повтор заданий поможет лучше уяснить пройденный материал.

Подготовка к сертификационному экзамену

В последней главе 30, "Подготовка к сертификационному экзамену", приведен перечень действий, которые стоит предпринять при окончательной подготовке к сдаче экзамена.

Другие особенности

Кроме основного содержимого каждой из глав, есть дополнительные учебные ресурсы, включая следующие.

- **Тренировочные тесты на веб-сайте**. На веб-странице книги `http://www.williamspublishing.com/Books/978-5-8459-1906-9.html` есть про-

граммное обеспечение Pearson IT Certification Practice Test для самотестирования. Имея образ DVD-диска и код цифрового ваучера для этой книги, запустите специальный экзамен, который очень похож на настоящий, как по курсу ICND1 и CCNA, так и по ICND2. (Ссылка на образ DVD-диска и инструкция по получению кода цифрового ваучера указана на веб-странице данной книги по адресу: `http://www.williamspublishing.com/Books/978-5-8459-1906-9.html`.)

- **Эмулятор CCNA Simulator Lite**. Эта "облегченная" версия популярного эмулятора CCNA Network Simulator от Pearson позволяет вам прямо сейчас проверить *интерфейс командной строки* (Command-Line Interface — CLI) Cisco. Нет никакой необходимости покупать реальное устройство или полнофункциональный эмулятор, чтобы приступить к изучению CLI. Просто установите его с образа DVD-диска, загруженного с веб-страницы книги.
- **Электронная книга.** Данное академическое издание укомплектовано бесплатным экземпляром на английском языке электронной книги и теста. Электронное издание представлено в трех форматах: PDF, EPUB и Mobi (исходный формат Kindle).
- **Видеоролики по расчету подсетей**. На образе DVD-диска также есть специальные видеоролики, помогающие понять принципы IP-адресации и методы расчета подсетей, в частности, как использовать методы расчета, описанные в этой книге.
- **Упражнения по расчету подсетей**. В приложениях, которые можно загрузить с веб-страницы книги, есть большой набор упражнений, соответствующих главам книги. Каждое приложение содержит набор задач по расчету подсетей с решениями для каждого упражнения и объяснениями того, как эти решения найдены. Это отличный ресурс для того, чтобы лучше и быстрее разобраться в принципах и методах расчета подсетей.
- **Дополнительные упражнения**. На веб-странице книги содержатся также еще несколько приложений с другими практическими задачами по отдельным главам книги. Используйте их для закрепления навыков практической работы.
- **Видеолекция.** Образ DVD-диска включает еще четыре видеоролика на английском языке по темам основ коммутаторов, навигации CLI, конфигурации маршрутизаторов и сетям VLAN.
- **Дополнительные материалы на веб-сайте**. На веб-сайте `www.ciscopress.com/title/9781587144851` представлены дополнительные материалы и обновления, которые появились в экзамене с момента выхода книги. Читатель может периодически заходить по указанному адресу и просматривать обновления, которые предоставляет автор книги, а также дополнительные материалы для подготовки к экзамену.
- **Веб-сайт** `www.pearsonitcertification.com`. Это великолепный ресурс по всем темам, связанным с сертификацией IT. Обратитесь к великолепным статьям, видео, блогам и другим средствам подготовки к сертификации CCNA Routing and Switching от лучших авторов и профессиональных преподавателей.

- **Симулятор CCNA Simulator.** Если вы ищете более профессиональный практикум, то можете рассмотреть возможность покупки эмулятора CCNA Network Simulator. Вы можете купить экземпляр этого программного обеспечения от Pearson по адресу `http://pearsonitcertification.com/networksimulator` или в другом месте. Чтобы помочь вам в изучении, я написал руководство, которое сопоставляет каждую из этих лабораторных работ в эмуляторе с определенным разделом данной книги. Вы можете получить это руководство бесплатно на вкладке "Extras" веб-сайта поддержки.
- **Веб-сайт автора и его блоги.** Автор поддерживает веб-сайт, содержащий инструментальные средства и ссылки, полезные при подготовке к экзаменам CCENT и CCNA Routing and Switching. Сайт предоставляет информацию, которая поможет вам создать собственную лабораторную работу, исследовать соответствующие страницы по каждой главе этой книги и книги по ICND2, а также блоги автора CCENT Skills и CCNA Skills. Начните с адреса `www.certskills.com`, а затем переходите на интересующие вас вкладки.

Структура книги, главы и приложения

Книга состоит из 29 основных глав, в каждой из которых рассмотрен определенный набор тем экзамена ICND1. В последней главе представлено краткое резюме по материалам книги и даны советы по сдаче сертификационного экзамена. Краткое описание глав приведено ниже.

Часть I "Основы сетей"

- **Глава 1, "Сетевые модели TCP/IP и OSI".** Знакомит с терминологией, обусловленной наличием двух различных сетевых архитектур, а именно TCP/IP и OSI.
- **Глава 2, "Основы сетей LAN".** Посвящена концепциям и терминологии наиболее популярной технологии физического и канального уровней локальных сетей — Ethernet.
- **Глава 3, "Основы сетей WAN".** Посвящена концепциям и терминологии наиболее распространенных технологий канального уровня распределенных сетей (WAN), а именно протоколу HDLC
- **Глава 4, "Основы IPv4-адресации и маршрутизации".** Посвящена основному протоколу сетевого уровня модели TCP/IP — протоколу Интернета (IP). В ней описаны основы IP-технологий (IPv4), в частности IPv4-адресация и маршрутизация.
- **Глава 5, "Основы протокола TCP/IP: передача данных и приложения".** Содержит введение в основы двух главных протоколов транспортного уровня модели TCP/IP — протокола TCP и протокола пересылки дейтаграмм, UDP.

Часть II "Коммутация в локальных сетях"

- **Глава 6, "Построение локальных сетей на базе коммутаторов".** Содержит углубленное и расширенное описание технологий локальных сетей, представленных в главе 2, в частности, обсуждаются роль и функции коммутатора LAN.

- **Глава 7, "Работа с коммутаторами Cisco".** Описаны методы подключения, проверки и настройки коммутаторов Catalyst компании Cisco.
- **Глава 8, "Настройка коммутаторов Ethernet".** Посвящена описанию функций коммутаторов: настройкам скорости и дуплексности портов, защите портов, методам обеспечения безопасности интерфейса командной строки и настройкам IP-адреса коммутатора.
- **Глава 9, "Реализация виртуальных локальных сетей".** Описаны концепции и конфигурации виртуальных локальных сетей, включая магистральное соединение VLAN и протокол создания магистралей VLAN.
- **Глава 10, "Поиск и устранение неисправностей на коммутаторах Ethernet".** Посвящена методам проверки работы коммутирующих устройств, преимущественно с помощью команд группы `show`.

Часть III "IPv4-адресация и создание подсетей"

- **Глава 11, "Перспективы создания подсетей IPv4".** Рассматриваются все концепции создания подсетей, начиная с классовой сети (A, B или C) и включая анализ требований, выбор, расчет подсети, перенос результата на бумагу, а также всю подготовку к установке, настройку устройств и использование подсети.
- **Глава 12, "Анализ классовых сетей IPv4".** Первоначально IPv4-адреса относились к нескольким классам при одноадресатных IP-адресах, начинающихся с классов A, B и C. В данной главе исследуется все, связанное с классами адресов, и концепции сети IP, порожденные этими классами.
- **Глава 13, "Анализ существующих масок подсети".** В большинстве случаев кто-то уже успел поработать перед вами и установить в сети маску подсети. Что это означает? Что дает эта маска? В данной главе речь идет о том, как по маске (и сети IP) выяснить такие ключевые факты, как размер подсети (количество хостов) и количество подсетей в сети.
- **Глава 14, "Анализ существующих подсетей".** Поиск и устранение большинства проблем подключения начинается с выяснения IP-адреса и маски. В этой главе рассматривается поиск упомянутой пары и демонстрируется, как осматривать и анализировать подсеть, в которой располагается IP-адрес, включая выяснение идентификатора подсети, диапазона адресов в подсети и широковещательного адреса подсети.

Часть IV "Реализация IP-адресации версии 4"

- **Глава 15, "Работа с маршрутизаторами Cisco".** Очень похожа на главу 8, но только посвящена маршрутизаторам, а не коммутаторам.
- **Глава 16, "Настройка IPv4-адресов и маршрутов".** Рассматриваются добавление в конфигурацию интерфейса маршрутизатора IPv4-адреса, а также маршруты, создаваемые маршрутизатором в результате настройки статических маршрутов IPv4.
- **Глава 17, "Самообучение маршрутов IPv4 с использованием OSPFv2".** Объясняется взаимодействие маршрутизаторов при поиске всех наилучших маршрутов к каждой подсети с использованием протокола маршрутизации. Опи-

сана также настройка протокола маршрутизации OSPF для использования с протоколом IPv4.

- **Глава 18, "Настройка и проверка подключения хостов".** Обсуждаются некоторые из инструментальных средств, используемых при настройке протокола IPv4 на хостах. В частности, обсуждаются такие средства, как DHCP, `ping` и `traceroute`, а также настройка параметров IPv4 на хосте.

Часть V "Дополнительные концепции IPv4-адресации"

- **Глава 19, "Проект подсети".** Прямо противоположный подход к созданию подсетей IPv4 по сравнению с приведенным в части III. В отличие от него, в данной главе рассматриваются вопросы о том, почему могла быть выбрана конкретная маска, и если она выбрана, то какие идентификаторы существуют в подсети.
- **Глава 20, "Маски подсети переменной длины".** Переводит создание подсетей IPv4 на другой уровень сложности, когда разные подсети в той же сети могут использовать разные маски подсети, чтобы у подсетей в той же сети были разные размеры.
- **Глава 21, "Суммирование маршрутов".** Описывается процесс, позволяющий настроить протокол маршрутизации так, чтобы он анонсировал один маршрут для большого набора адресов, а не множество маршрутов для каждого меньшего набора.

Часть VI "Службы IPv4"

- **Глава 22, "Простые списки управления доступом IPv4".** Описано, как стандартный список ACL позволяет фильтровать пакеты на основании IP-адреса отправителя, чтобы маршрутизатор не передавал их.
- **Глава 23, "Расширенные списки управления доступом и защита устройств".** Описаны именованные и нумерованные списки ACL. Основное внимание уделено тому, как расширенный список ACL может распознавать пакеты на основании IP-адреса отправителя или получателя, а также распознавать номера портов TCP и UDP отправителя или получателя.
- **Глава 24, "Трансляция сетевых адресов".** Подробно рассматриваются концепции, лежащие в основе исчерпания пространства IPv4-адресов, а также то, как технология NAT позволяет решить эту проблему при помощи трансляции адресов портов (PAT). Кроме того, демонстрируется также настройка NAT на маршрутизаторах при помощи интерфейса командной строки операционной системы Cisco IOS.

Часть VII "Протокол IP версии 6"

- **Глава 25, "Основы протокола IP версии 6".** Обсуждаются фундаментальные концепции протокола IP версии 6, при этом основное внимание уделяется правилам выбора и интерпретации IPv6-адресов.
- **Глава 26, "IPv6-адресация и создание подсетей".** Рассматривает две ветви одноадресатных IPv6-адресов: глобальных одноадресатных адресов и уникальных локальных адресов, действующих наподобие открытых и частных IPv4-адресов соответственно. Помимо этого, демонстрируется создание подсетей IPv6.

- **Глава 27, "Реализация IPv6-адресации на маршрутизаторах".** Демонстрируется настройка маршрутизации IPv6 и адресации на маршрутизаторах. Обсуждаются также локальные одноадресатные адреса и другие специальные адреса, используемые маршрутизаторами.
- **Глава 28, "Реализация IPv6-адресации на хостах".** Демонстрируется настройка конфигурации IPv6 на хостах, с акцентом на двух методах, позволяющих хостам изучать параметры протокола IPv6: протокол DHCPv6 с фиксацией состояния и автоматическая настройка адресов без фиксации состояния (Stateless Address Autoconfiguration — SLAAC).
- **Глава 29, "Реализация маршрутизации по протоколу IPv6".** Рассматривается добавление маршрутов в таблицу маршрутизации маршрутизатора как при помощи статической конфигурации, так и протокола OSPF версии 3 (OSPFv3).

Часть VIII "Подготовка к экзамену"

- **Глава 30, "Подготовка к сертификационному экзамену".** Содержит план окончательной подготовки к сертификационному экзамену после изучения книги, включая дополнительные материалы и ключевые моменты.

Часть IX "Приложения"

- **Приложение А, "Справочные числовые таблицы".** Состоит из нескольких таблиц с цифровой информацией, включая таблицу преобразования чисел в двоичную систему и список степеней числа 2.
- **Приложение Б, "Обновление экзамена ICND1".** Состоит из небольших тем и блоков материала для повторения пройденных тем. Это приложение время от времени обновляется и размещается по адресу `www.ciscopress.com/title/1587143852`. Материалы, доступные на момент издания книги, были добавлены в это приложение. Здесь также приведена подробная инструкция о том, как загрузить наиболее свежую версию этого приложения.
- **Список терминов.** содержит определения всех терминов из разделов "Ключевые термины", приведенных в конце каждой главы.

Часть X "Приложения (на веб-сайте)"

Перечисленные ниже приложения в цифровом формате размещены на веб-странице книги `http://www.williamspublishing.com/Books/978-5-8459-1906-9.html`.

- **Приложение В, "Ответы на контрольные вопросы".** Содержит ответы на контрольные вопросы всех глав.
- **Приложение Г, "Практические задания главы 12. Анализ классовых сетей IPv4".** Содержит список практических задач, связанных с материалом главы 12. В частности, задачи о выяснении адреса классовой сети, в которой располагается адрес, и всех других фактов об этой сети.
- **Приложение Д, "Практические задания главы 13. Анализ существующих масок подсети".** Содержит список практических задач, связанных с материалом главы 13. В частности, задачи на преобразование между тремя форматами масок, исследование существующих масок, выявление структуры IP-адресов, вычисление количества подсетей и хостов на подсеть.

- **Приложение Е, "Практические задания главы 14. Анализ существующих подсетей".** Содержит список практических задач, связанных с материалом главы 14. В частности, вопросы и практические задачи на выяснение IP-адресов и масок, поиск идентификаторов подсети, широковещательных адресов и диапазонов IP-адресов в подсети.
- **Приложение Ж, "Практические задания главы 19. Проект подсети".** Содержит список практических задач, связанных с материалом главы 19. В частности, вопросы и практические задачи на исследование набора требований, определение масок, соответствующих этим требованиям (если нужно), и выбор наилучшей из них на основании предпочтений. Кроме того, поиск всех идентификаторов подсети в классовой сети, когда задана единая маска, используемая по всей сети.
- **Приложение З, "Практические задания главы 20. Маски подсети переменной длины".** Содержит список практических задач, связанных с материалом главы 20, включая задачи на поиск места для добавления новой подсети VLSM без перекрытия.
- **Приложение И, "Практические задания главы 21. Суммирование маршрутов".** Содержит список практических задач, связанных с материалом главы 21. В частности, вопросы и практические задачи на поиск наилучшего суммарного маршрута, включающего в список все подсети.
- **Приложение К, "Практические задания главы 22. Простые списки управления доступом IPv4".** Содержит список практических задач, связанных с материалом главы 22. В частности, вопросы и практические задачи, позволяющие опробовать на практике работу с шаблонами масок ACL.
- **Приложение Л, "Практические задания главы 25. Основы протокола IP версии 6".** Содержит список практических задач, связанных с материалом главы 25. В частности, практические задачи по сокращению полных и развертыванию сокращенных IPv6-адресов.
- **Приложение М, "Практические задания главы 27. Реализация IPv6-адресации на маршрутизаторах".** Содержит список практических задач, связанных с материалом главы 27. В частности, практику использования процесса EUI-64 для создания IPv6-адреса, а также поиска требуемого многоадресатного узла на основании одноадресатного адреса.
- **Приложение Н, "Таблицы для запоминания материала".** Содержит ключевые таблицы и списки из всех глав, в которых удалена некоторая информация. Эти таблицы можно распечатать и использовать для тренировки памяти — заполнить их, не заглядывая в книгу. Их цель помочь запомнить те факты, которые могут быть полезны на экзаменах.
- **Приложение О, "Таблицы для запоминания материала с ответами".** Содержит заполненные таблицы (т.е. фактически ответы) к приложению Н.
- **Приложение П, "Решения для диаграмм связей".** Содержит рисунки с ответами на все упражнения с диаграммами связей.
- **Приложение Р, "План изучения".** Таблица с основными этапами, по которой можно проследить свой прогресс в обучении.

Справочная информация

Этот короткий раздел содержит несколько тем, доступных по ссылке из других мест в книге. Их можно прочитать сразу, а можно пропустить и вернуться к ним позже. В частности, обратите внимание на заключительную страницу введения, на которой приведена контактная информация и указан способ связи с издательством.

Установка процессора Pearson IT Certification Practice Test и вопросов

Расположенный на веб-странице книги образ DVD-диска содержит экзаменационный процессор Pearson IT Certification Practice Test (PCPT), позволяющий оценить свои знания на реалистичных экзаменационных вопросах и тестлетах. Используя процессор Pearson IT Certification Practice Test, можно учиться, находясь в режиме обучения, или смоделировать реальные условия экзамена ICND1 или CCNA.

Процесс установки состоит из двух основных этапов. Сам экземпляр процессора Pearson IT Certification Practice Test содержится на образе DVD-диска, но там нет базы данных экзаменационных вопросов ICND1 и CCNA. После установки программного обеспечения PCPT его последнюю версию, а также базы данных с вопросами можно загрузить по Интернету.

Используйте цифровой ваучер для доступа к электронным версиям книги и экзаменационным вопросам

Для использования экзаменационного программного обеспечения следует задействовать цифровой ваучер продукта (инструкция по получению цифрового ваучера приведена на веб-странице книги по адресу: `http://www.williamspublishing.com/Books/978-5-8459-1906-9.html`). Для этого необходимо предпринять следующее.

Этап 1 Если у вас уже есть учетная запись Cisco Press, перейдите на сайт `www.ciscopress.com/account` и зарегистрируйтесь. Если учетной записи нет, перейдите по адресу `www.ciscopress.com/join` и создайте ее

Этап 2 На странице учетной записи найдите поле Digital Product Voucher вверху правого столбца

Этап 3 Введите свой код цифрового ваучера и щелкните на кнопке Submit (Передать)

ВНИМАНИЕ!

Цифровой ваучер предназначен для одноразового использования, не перевадайте его третьим лицам!

Этап 4 Теперь на странице вашей учетной записи в разделе покупок появились ссылки на товары и загрузки, а также информация о коде доступа (Access Code) к экзаменационным вопросам. Для загрузки файлов электронной книги щелкните на ссылках. Для доступа и загрузки экзаменационных вопросов Premium Edition к процессору Pearson IT Certification Practice Test используйте код доступа, как описано в следующих разделах

Установка программного обеспечения с образа DVD-диска

Процесс установки данного программного обеспечения весьма прост по сравнению с установкой другого программного обеспечения. Если программное обеспечение Pearson IT Certification Practice Test от другого продукта компании Pearson уже установлено, нет никакой необходимости устанавливать его повторно. Просто запустите его на своем рабочем столе и перейдите к активации экзаменов из этой книги, используя код доступа (см. предыдущий раздел). Ниже приведена последовательность действий по установке.

Этап 1 Смонтируйте обаз DVD-диска в вашей операционной системе. За инструкциями обратитесь к поисковой системе

Этап 2 Программное обеспечение будет запущено автоматически. Оно позволит получить доступ ко всему программному обеспечению Cisco Press на виртуальном DVD-диске, включая экзаменационный процессор и приложения к книге на английском языке (скачать эти же приложения на русском языке можно по ссылке, которая приведена на веб-странице книги `http://www.williamspublishing.com/Books/978-5-8459-1906-9.html`). В главном меню щелкните на ссылке Install the Exam Engine (Установить экзаменационный процессор)

Этап 3 Отвечайте на вопросы в окнах мастера установки, как и при установке любого программного обеспечения

Процесс установки позволяет активировать экзамены при помощи кода доступа. Этот процесс требует регистрации на веб-сайте Pearson. Поскольку регистрация необходима для активации экзамена, пожалуйста, зарегистрируйтесь, когда вас попросят. Если регистрация на веб-сайте Pearson уже есть, повторная регистрация не нужна. Просто используйте свою уже существующую учетную запись.

Активация и загрузка экзаменационных вопросов

После установки экзаменационного процессора необходимо активировать связанные с этой книгой экзаменационные вопросы (если это еще не было сделано в процессе установки) следующим образом.

Этап 1 Запустите программное обеспечение PCPT из меню кнопки Start (Пуск) операционной системы Windows или при помощи пиктограммы на рабочем столе

Этап 2 Для активации и загрузки связанных с этой книгой экзаменационных вопросов на вкладке My Products или Tools щелкните на кнопке Activate

Этап 3 На следующем экране введите код доступа, указанный в продуктах Premium Edition на странице вашей учетной записи на сайте `www.ciscopress.com`. Затем щелкните на кнопке Activate

Этап 4 Процесс активации загрузит экзамен. Щелкните на кнопке Next, а затем на Finish

По завершении процесса активации на вкладке My Products должен быть указан ваш новый экзамен. Если экзамен не виден, удостоверьтесь, что перешли в меню на вкладку My Products. Теперь программное обеспечение и экзамен практически готовы к использованию. Выберите экзамен и щелкните на кнопке Open Exam.

Для обновления уже активированного и загруженного экзамена перейдите на вкладку Tools, а затем щелкните на кнопке Update Products. Обновление экзаменов гарантирует наличие последних изменений и обновлений данных экзамена.

Если необходимо проверить обновления к программному обеспечению PCPT, перейдите на вкладку Tools и щелкните на кнопке Update Application. Это гарантирует наличие последней версии программного обеспечения.

Экзаменационные базы данных PCPT этой книги

Экзаменационные вопросы поставляются в различных экзаменах или экзаменационных базах данных. При установке программного обеспечения PCPT и вводе кода доступа загружается последняя версия всех экзаменационных баз данных. Только по одной книге ICND1 вы получаете 10 разных экзаменов или 10 разных наборов вопросов (рис. 1.2).

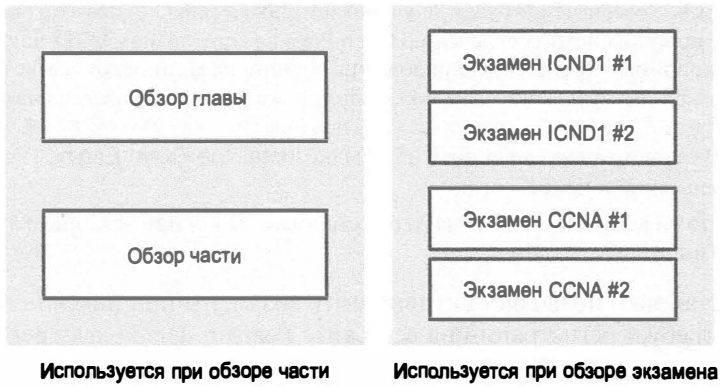


Рис. 1.2. Экзамены, экзаменационные базы данных PCPT и время их использования

Любую из этих баз данных можно использовать в любое время как в режиме обучения, так и в режиме экзаменационной практики. Однако многие предпочитают отложить некоторые из экзаменов до завершения изучения всей книги. На рис. 1.2 показан приведенный ниже план.

- Во время обзора части используйте процессор PCPT для обзора вопросов глав (в приложении обозначено как "Book Questions") данной части в режиме обучения.
- Во время обзора части используйте вопросы, специально предназначенные для данной части книги (вопросы в обзоре части) в режиме обучения.
- Оставьте экзамены для использования с заключительной главой книги, используя режим имитации экзамена, как описано в главе 30.

Эти два режима PCPT обеспечивают более удобный способ обучения по сравнению с реальным экзаменом, где время ограничено. В режиме обучения ответы можно просмотреть немедленно, что облегчает изучение тем. Кроме того, в базе данных можно выбрать некое подмножество вопросов, например, можно просмотреть вопросы только глав из одной части книги.

Режим экзамена практически имитирует фактический экзамен. Он выдает набор вопросов по всем главам и требует ответить на них за установленное время. По завершении предоставляются результаты экзамена.

Как просмотреть вопросы только обзоров глав конкретной части

Каждый обзор части содержит повтор вопросов из обзоров глав этой части. Хотя вполне можно пролистать страницы книги и найти вопросы всех обзоров глав, значительно проще просмотреть эти вопросы в приложении PCPT, достаточно немного попрактиковаться в чтении вопросов на экзаменационном программном обеспечении. Но их можно прочитать и в книге.

Для просмотра вопросов обзора главы в приложении PCPT необходимо выбрать в меню пункт Book Questions (Вопросы из книги) и главы соответствующей части. Это можно сделать так.

Этап 1 Запустите программное обеспечение PCPT

Этап 2 В главном меню выберите элемент данного продукта по имени Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide и щелкните на Open Exam

Этап 3 Вверху следующего окна должны быть перечислены экзамены; установите флажок напротив ICND1 Book Questions и сбросьте другие флажки. Затем выберите вопросы "book", т.е. вопросы из обзоров в конце каждой главы

Этап 4 В этом же окне можно щелкнуть внизу экрана, чтобы сбросить все главы, а затем выбрать все главы необходимой части книги

Этап 5 Справа в окне выберите другие параметры

Этап 6 Для запуска набора вопросов щелкните на кнопке Start

Как просмотреть вопросы только обзоров частей

Среди предоставляемых этой книгой баз данных экзаменационных вопросов есть база, созданная исключительно для изучения обзоров частей. Вопросы в обзорах глав сосредоточены больше на фактах и простых приложениях. Вопросы в обзорах частей, напротив, больше похожи на реальные экзаменационные вопросы.

Для просмотра этих вопросов следуйте той же инструкции, что и при просмотре вопросов из обзоров глав, но вместо базы данных "Book" выбирайте базу "Part Review".

Этап 1 Запустите программное обеспечение PCPT

Этап 2 В главном меню выберите элемент данного продукта по имени Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide и щелкните Open Exam

Этап 3 Вверху следующего окна должны быть перечислены экзамены; установите флажок напротив Part Review Questions и сбросьте другие флажки. В результате будут выбраны вопросы из обзора в конце части

Этап 4 В этом же окне можно щелкнуть внизу экрана, чтобы сбросить все задачи, а затем выбрать вопросы из необходимых обзоров частей. В результате приложение PCPT загрузит вопросы из выбранных обзоров частей

Этап 5 Справа в окне выберите другие параметры

Этап 6 Для запуска набора вопросов щелкните на Start

О диаграммах связей

Диаграммы связей — это многоцелевой организационный графический инструмент. Например, диаграммы связей применяются как альтернативный способ делать заметки.

Диаграммы связей могут также использоваться для улучшения осознания концепций. Они подчеркивают взаимосвязи и отношения между понятиями. Уделяя время обдумыванию изучаемой темы и организуя диаграмму связей, вы укрепляете существующие и создаете новые ассоциации в памяти, а также вырабатываете собственную систему взглядов.

Короче говоря, диаграммы связей помогут усвоить то, что вы изучаете.

Механика диаграмм связей

Каждая диаграмма связей начинается с чистого листа бумаги или окна в графическом приложении. Сначала изображается большая центральная идея с ветвями, распространяющимися в любом направлении. Ветви содержат меньшие концепции, идеи, команды, изображения, т.е. все, что должна представлять идея. Все концепции, которые могут быть сгруппированы, должны быть помещены рядом. При необходимости можно создавать все более и более глубокие ветви, хотя большинство диаграмм связей в этой книге не будет превышать лишь нескольких уровней.

ВНИМАНИЕ!

Хотя о диаграммах связей написано множество книг, Тони Бузан (Tony Buzan) продолжает формализацию и популяризацию диаграмм связей. Более подробная информация о диаграммах связей приведена на его веб-сайте по адресу `www.thinkbuzan.com`.

На рис. I.3 приведен пример диаграммы связей, отображающей часть концепций IPv6-адресации из части VII книги. Центральная концепция диаграммы связей — IPv6-адресация, а обзор части требует обдумать все факты, относящиеся к IPv6-адресации, и организовать их в диаграмму связей. Диаграмма связей позволяет наглядней представить концепции по сравнению с их текстовым описанием.

Рис. I.3. Пример диаграммы связей

О диаграммах связей, используемых в обзорах частей

В обзорах частей этой книги предлагаются упражнения с диаграммами связей. В этом коротком разделе перечислены некоторые из подробностей об упражнениях с диаграммами связей, собранные в одном месте.

Разделы обзоров частей используют два основных вида упражнений с диаграммами связей.

- Упражнения на конфигурацию требуют вспомнить взаимосвязанные команды конфигурации и сгруппировать их. Например, связанные команды в упражнении на конфигурацию, являющиеся подкомандами интерфейса, должны быть сгруппированы, но, как показано, в режиме конфигурации внутреннего интерфейса.
- Упражнения по проверке требуют обдумать вывод команд `show` и связать вывод либо с влияющими на него командами конфигурации, либо с концепциями, объясняющими значение данной части вывода.

Конфигурационные диаграммы связей можно создать на бумаге либо с помощью любого подходящего программного обеспечения или любого графического редактора. Существует также множество специализированных приложений диаграмм связей. Независимо от способа рисования диаграммы связей должны подчиняться следующим правилам.

- Если времени для этого упражнения мало, сэкономьте его, составив собственную диаграмму связей, а не смотрите в предложенные ответы. Обучение происходит при самостоятельном решении задачи и создании собственной диаграммы связей.
- Закройте книгу, все свои заметки и не подглядывайте в них при первом создании диаграмм. Проверьте, получится ли нарисовать их без книги, своих заметок, Google и другой помощи.
- Прежде чем заглянуть в свои заметки, пройдите все диаграммы связей, заданные в обзоре части.
- Просмотрите свои заметки, чтобы завершить все диаграммы связей.
- Получая результаты, делайте заметки, чтобы использовать их впоследствии при окончательной подготовке к экзамену.

И наконец, при обучении с использованием этих средств учтите еще две важные рекомендации. Во-первых, используйте поменьше слов для каждого узла в диаграмме связей. Следует запомнить саму концепцию и ее взаимосвязи, а не объяснить идею кому-то еще. Пишите только то, что помните о концепции. Во-вторых, если работа с диаграммами связей вам не подходит, откажитесь от них. Делайте вместо них просто заметки на листе бумаги. Попытайтесь выполнить важнейшую часть упражнения, размышление над взаимодействием концепций, не позволяя инструменту мешать вам.

О приобретении практических навыков

Для сдачи экзамена нужны практические навыки использования маршрутизаторов и коммутаторов Cisco, а именно работы с интерфейсом командной строки Cisco (Command-Line Interface — CLI). CLI Cisco — это текстовый пользовательский интерфейс команд и ответов, позволяющий ввести команду для устройства (маршрутизатора или коммутатора) и получить ответное сообщение. Для ответов на экзаменационные вопросы с симлетами необходимо знать множество команд и быть в состоянии переходить в нужное место интерфейса CLI, чтобы использовать эти команды.

Наилучший способ овладеть этими командами — использовать их на практике. При первом чтении части I этой книги необходимо решить, как вы планируете приобретать навыки в CLI. В следующем разделе обсуждаются возможности и средства приобретения практических навыков работы с CLI.

Возможности лабораторных работ

Для эффективной выработки практических навыков работы с CLI нужны либо реальные маршрутизаторы и коммутаторы, либо, по крайней мере, нечто, действующее, как они. Новички в технологиях Cisco обычно предпочитают другие возможности для приобретения этих навыков.

В первую очередь можно использовать реальные маршрутизаторы и коммутаторы Cisco. Можно купить новые или подержанные либо позаимствовать на работе, их можно также взять на прокат. Можно даже арендовать виртуальный маршрутизатор или коммутатор Cisco для лабораторных работ от Cisco Learning Labs.

Эмуляторы предоставляют и другую возможность. Эмуляторы маршрутизаторов и коммутаторов — это программные продукты, подражающие поведению интерфейса CLI Cisco, как правило, в учебных целях. У этих продуктов есть дополнительное преимущество при обучении: они комплектуются упражнениями и лабораторными работами.

Эмуляторы бывают всех форм и размеров, но издатель предлагает эмуляторы, специально разработанные для помощи в подготовке к экзаменам CCENT и CCNA, а кроме того, они соответствуют этой книге! Эмуляторы Pearson CCENT Network Simulator и Pearson CCNA Network Simulator обеспечивают превосходную среду для практики ввода команд, а также предоставляют сотни специализированных лабораторных работ, призванных помочь подготовиться к экзамену. Базовый код у обоих продуктов одинаков. Просто продукт CCNA включает лабораторные работы и для ICND1, и для ICND2, в то время как продукт CCENT — только лабораторные работы для ICND1.

Автор книги вовсе не указывает вам, какие средства использовать, но вам так или иначе придется спланировать, как получать профессиональные навыки. Просто достаточно знать, что очень многие сдавшие экзамен при подготовке практиковались в использовании интерфейса CLI Cisco.

Я (Уэнделл) собрал на своем веб-сайте `certskills.com/labgear` некоторую информацию и мнения об этом решении. Эти страницы связаны с сайтами Dynamips и Pearson Simulator. Кроме того, поскольку данной информации нет ни в каком другом месте, этот веб-сайт содержит подробности создания лабораторных работ CCNA с использованием реальных маршрутизаторов и коммутаторов Cisco.

Коротко о Pearson Network Simulator Lite

Дискуссия о способе получения практических навыков может показаться сначала немного странной. Хорошая новость — у вас есть простой и бесплатный первый этап: книга укомплектована симулятором Pearson NetSim Lite.

Эта облегченная версия популярного эмулятора CCNA Network Simulator от Pearson позволяет прямо сейчас опробовать интерфейс командной строки Cisco (CLI). Нет никакой необходимости покупать реальное устройство или полнофунк-

циональный эмулятор, чтобы начать изучать интерфейс CLI. Достаточно установить его с образа DVD-диска.

Конечно, одна из причин наличия версии NetSim Lite на этом диске в том, что издатель надеется на покупку вами полной версии продукта. Но даже если вы не используете полную версию, то вполне можете использовать для обучения лабораторные работы версии NetSim Lite, а уже затем принимать решение о том, что использовать далее.

ВНИМАНИЕ!

Каждая из книг, ICND1 и ICND2, содержит разные версии продуктов Sim Lite с соответствующими лабораторными работами. Если вы купили обе книги, установите оба экземпляра продукта.

Дополнительная информация

Компания Cisco изредка может вносить изменения в программу, которые отражаются и в сертификационном экзамене CCNA Routing and Switching. Перед тем как сдавать соответствующие сертификационные экзамены, следует проверить, не изменились ли их темы, по адресам `www.cisco.com/go/ccna` и `www.cisco.com/go/ccent`.

Книга призвана помочь сетевому специалисту в обучении сетевым технологиям и сдаче сертификационных экзаменов CCENT и CCNA Routing and Switching. Эта книга — учебник от единственного авторизованного компанией Cisco издательства — Cisco Press. Издательство Cisco Press верит, что эта книга безусловно поможет читателю как в подготовке к экзамену CCNA, так и в практической работе. Мы надеемся, что вы с пользой проведете время за чтением этой книги.

Соглашения по синтаксису команд

Представленные ниже соглашения по синтаксису команд аналогичны соглашениям, используемым в *Справочнике по командам операционной системы IOS* (IOS Command Reference). В упомянутом справочнике используются следующие соглашения:

- **полужирным** шрифтом выделяются команды и ключевые слова, которые вводятся буквально, как показано в примерах реальной конфигурации и сообщений системы. Полужирным шрифтом выделяются команды, которые вводятся пользователем вручную (например, команда **show**);
- *курсивом* выделяются аргументы, для которых пользователь указывает реальные значения;
- с помощью вертикальной черты (|) разделяются альтернативные, взаимоисключающие элементы;
- в квадратных скобках ([ ]) указываются необязательные элементы;
- в фигурных скобках ({ }) указываются необходимые элементы;

- в фигурных скобках, помещенных в квадратные скобки `[{ }]`, указываются необходимые элементы в пределах необязательного элемента.

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш веб-сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши электронные адреса:

E-mail: `info@williamspublishing.com`
WWW: `http://www.williamspublishing.com`

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д.43, стр. 1
в Украине: 03150, Киев, а/я 152

Первые шаги

В этом разделе приведено несколько ценных советов по использованию данной книги для обучения. Уделите несколько минут чтению данного раздела, прежде чем переходить к главе 1, — это позволит извлечь больше пользы из изучения книги, независимо от того, используется ли она для подготовки к сертификационным экзаменам CCNA Routing and Switching или только для изучения базовых концепций работы с сетями.

Коротко о сертификационных экзаменах Cisco

Компания Cisco установила довольно высокую планку для сдачи экзаменов ICND1, ICND2 и CCNA. Любой может пройти обучение и сдать экзамен, но для этого недостаточно поверхностного чтения книги и наличия денег на оплату экзамена.

Сложность этих экзаменов обусловлена множеством аспектов. Каждый из экзаменов покрывает массу концепций, а также множество команд, специфических для устройств Cisco. Кроме знания, экзамены Cisco требуют также наличия навыков. Необходима способность анализировать и предсказывать происходящее в сети, а также способность правильно настраивать устройства Cisco для работы в этих сетях. Следует быть готовым к диагностике и устранению проблем, когда сеть работает неправильно.

Более сложные вопросы этих экзаменов напоминают мозаику, причем четырех фрагментов из каждых пяти, как правило, нет. Для решения задачи придется мысленно воссоздать недостающие части. Чтобы сделать это, нужно хорошо понимать все сетевые концепции и их взаимодействие. Следует также быть в состоянии сопоставить эти концепции с происходящим на устройстве и командами конфигурации, контролирующими данное устройство. Чтобы проанализировать сеть и установить, почему она теперь работает неправильно, понадобится сопоставить концепции и конфигурацию с выводом различных команд диагностики.

Например, тема создания подсетей IP подразумевает хорошее знание математических механизмов. Даже простой вопрос (слишком простой, чтобы быть реальным вопросом на экзамене) показывает, что для поиска идентификаторов подсети достаточно простого сложения и умножения.

Более реалистичный экзаменационный вопрос потребует для формулировки математической задачи объединения нескольких концепций. Например, в вопросе может быть дана схема сети, для которой требуется вычислить идентификатор подсети, используемый в указанной части схемы. Но на схеме нет никаких чисел вообще. Вместо них есть только вывод команды маршрутизатора, например команды `show ip ospf database`, которая действительно отображает некоторые числа. Однако, прежде чем эти числа можно будет использовать, возможно понадобится установить, как устройства настроены и что дали бы другие команды диагностики. Таким образом, вопрос будет выглядеть как головоломка на рис. 1. Части вопроса потребуется расставить по своим местам; это позволит, используя различные команды и применяя свои знания, найти другие части головоломки. В результате для данного вопроса останутся неизвестными только некоторые части.

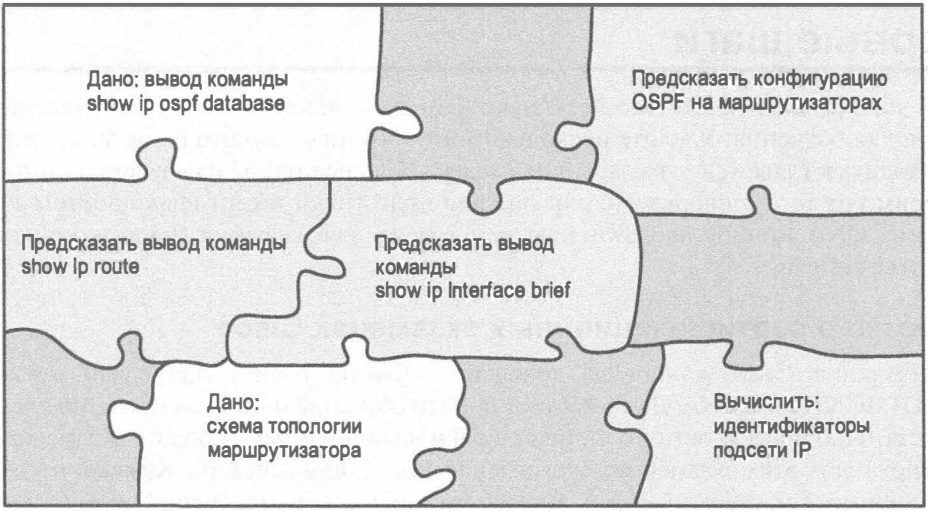


Рис. 1. Сборка головоломки требует аналитических навыков

Для приобретения таких навыков потребуется не только чтение и запоминание прочитанного. Конечно, в ходе обучения придется прочитать много страниц этой книги, узнать много фактов и запомнить взаимосвязь между ними. Однако бо́льшую часть книги составляет не текст для чтения, а упражнения, призванные помочь приобрести навыки для решения сетевых проблем.

Рекомендации по изучению книги

Если книга используется для изучения базовых сетевых концепций или подготовки к экзамену CCNA Routing and Switching, стоит обратить внимание на то, как именно использовать ее для достижения поставленной цели. Так что же необходимо кроме чтения и запоминания всех фактов для подготовки к сдаче экзамена CCNA Routing and Switching и успешной работы специалиста по сетям? Необходимо выработать навыки, уметь мысленно связать каждую концепцию с другими связанными с ней. Это потребует дополнительных усилий. Для помощи в этом на нескольких следующих страницах приведено пять ключевых точек зрения на то, как использовать эту книгу для приобретения этих навыков, прежде чем погрузиться в прекрасный, но сложный мир работы с сетями на базе устройств Cisco.

Не одна книга, а 29 коротких задач по чтению и проверке

Считайте свое обучение набором задач по чтению и проверке всех относительно небольших взаимосвязанных тем.

В среднем каждая из основных глав этой книги (1–29) насчитывает приблизительно по 22 страницы текста. При внимательном просмотре в начале любой из этих глав можно обнаружить раздел “Основные темы”. От него до раздела ”Обзор” в конце главы следует в среднем порядка 22 страниц.

Поэтому не считайте эту книгу одной большой книгой. Считайте задачу первого чтения главы отдельной задачей. Любой может прочитать 22 страницы — это не сложно. В каждой главе есть два или три основных раздела, которые вы можете чи-

тать только по одному в день. Либо выполните лабораторные работы главы или сделайте обзор того, что уже прочитано. Чтобы сделать содержимое этой книги более удобочитаемым и облегчить его изучение, оно организовано по темам небольшого размера.

Не пренебрегайте практическими заданиями глав

Планируйте выполнение задач раздела "Обзор" в конце каждой главы.

Каждая глава завершается разделом "Обзор" с практическими задачами и упражнениями на повторение материала. Выполнение этих задач и упражнений в конце главы действительно помогает подготовиться. Не откладывайте решение этих задач! Раздел "Обзор" в конце главы поможет с первой фазой углубления знаний и приобретения навыков по ключевым темам, поможет запомнить термины и увязать концепции в памяти так, чтобы вспомнить их в соответствующий момент.

В разделах "Обзор" каждой главы, как правило, содержатся следующие темы.

- Резюме
- Контрольные вопросы
- Ключевые темы
- Заполните таблицы и списки по памяти
- Ключевые термины
- Таблицы команд
- Обзор конфигурации
- Упражнения по созданию подсетей

Используйте части книги как основные этапы

Рассматривайте книгу как семь основных этапов, по одному на каждую главную тему.

Кроме вполне очевидной организации по главам, эта книга объединяет главы в семи основных темах, соответствующих частям книги. Завершение каждой части означает конец изучения одной из областей знаний. Уделите концу каждой части дополнительное время. Решите в конце каждой части задачи раздела "Обзор части". Выясните свои слабые и сильные стороны. Семь частей этой книги представлены на рис. 2.

Задачи раздела "Обзор части" призваны помочь применить изученные в данной части концепции в условиях экзамена. В некоторых заданиях приведены примеры простых вопросов, чтобы можно было продумать и проанализировать задачу. Этот процесс поможет усовершенствовать свои знания и понять то, что было не понято до конца. В других заданиях используются упражнения, требующие мысленно объединить теоретические концепции с командами настройки и диагностики. Все задания раздела "Обзор части" как раз и помогут выработать необходимые навыки.

Обратите внимание, что для доступа к практическим заданиям в обзорах частей требуется использовать программное обеспечение Pearson Certification Practice Test (PCPT). В обзорах частей требуется также повторять вопросы из обзоров глав, но с помощью программного обеспечения PCPT. Каждый обзор части требует также доступа к определенному набору вопросов, предназначенных для обзора концепций дан-

ной части. Обратите внимание, что программное обеспечение PCPT и базы данных экзаменационных вопросов, предоставляемые с этой книгой, дают право и на дополнительные вопросы; в главе 30 приведены дополнительные рекомендации о том, как лучше всего использовать эти вопросы для окончательной подготовки к экзамену.

Семь основных этапов. Части книги

Основы сетей	Задачи части
Коммутация в локальных сетях	Задачи части
IPv4-адресация и создание подсетей	Задачи части
Реализация IP-адресации версии 4	Задачи части
Дополнительные концепции IPv4-адресации	Задачи части
Службы IPv4	Задачи части
Протокол IP версии 6	Задачи части

Рис. 2. Части книги как основные этапы

Используйте главу для окончательной подготовки для совершенствования навыков

Выполняйте задачи, вынесенные в заключительную главу книги.

У заключительной главы две главные цели. Во-первых, она поможет углубить аналитические навыки, необходимые для ответа на более сложные вопросы экзамена. Многие вопросы требуют объединения понимания концепций со знанием конфигурации, проверки и диагностики. Простого чтения недостаточно для приобретения таких навыков, а задачи данной главы окажут в этом помощь.

Задачи заключительной главы помогут также выявить свои слабые стороны. Это позволит подготовиться к сложным вопросам на экзамене и выявить любые пробелы в знаниях. Большинство вопросов специально разработано так, чтобы выявить наиболее распространенные ошибки и заблуждения, а также помочь избежать части затруднений, с которыми обычно сталкиваются на реальном экзамене.

Установите цели и следите за прогрессом

И наконец, прежде чем читать книгу и выполнять учебные задачи, уделите время выработке плана, установке неких целей и подготовьтесь к отслеживанию своего прогресса.

Создание списков задач может быть полезно, а может, и нет, в зависимости от индивидуальных особенностей, но выбор целей поможет всем, а для этого необходимо знать, какую работу предстоит выполнить.

Что касается списка выполняемых при обучении задач, то не стоит его слишком детализировать. (В список можно включить все задачи из раздела "Обзор" в конце каждой главы, задачи из всех разделов "Обзор части" и задачи из заключительной главы.) Вполне достаточно списка лишь основных задач.

Для каждой обычной главы следует отследить по крайней мере две задачи: чтение раздела "Основные темы" и выполнение заданий раздела "Обзор" в конце главы. И не забывайте, конечно, задачи разделов "Обзор части" и заключительной главы. Пример плана для первой части книги приведен в табл. 1.

Таблица 1. Пример выдержки из плана

Элемент	Задача	Дата	Первая дата завершения	Вторая дата завершения (опционально)
Глава 1	Прочитать основные темы			
Глава 1	Выполнить задания обзора			
Глава 2	Прочитать основные темы			
Глава 2	Выполнить задания обзора			
Глава 3	Прочитать основные темы			
Глава 3	Выполнить задания обзора			
Глава 4	Прочитать основные темы			
Глава 4	Выполнить задания обзора			
Глава 5	Прочитать основные темы			
Глава 5	Выполнить задания обзора			
Обзор части I	Выполнить задания обзора части			

ВНИМАНИЕ!

Приложение P, "План изучения", на веб-сайте содержит полный план в виде таблицы. Эту таблицу можно изменить и сохранить в файле, чтобы отслеживать даты выполнения поставленных задач.

Используйте даты только как способ контроля за процессом обучения, а не как последний срок, к которому обязательно нужно успеть. Выбирайте реальные сроки, в которые можно уложиться. Устанавливая свои цели, учитывайте скорость чтения и объемы раздела основных тем каждой главы (его можно выяснить в содержании). Если закончите задачу быстрее, чем запланировано, можете сдвинуть следующие даты.

Если пропустите несколько дат, не расстраивайтесь и не пропускайте задачи в концы глав! Вместо этого подумайте о том, как скорректировать свои цели или немного плотнее поработать над обучением.

Дополнительные задания перед началом

Перед началом придется выполнить еще несколько дополнительных заданий: установить программное обеспечение, найти файлы PDF и т.д. Эти задания можно выполнить сейчас или когда появится перерыв в изучении первых глав книги. Но сделайте это пораньше, чтобы в случае проблем с установкой не останавливать изучения до момента их устранения.

Зарегистрируйтесь (бесплатно) в учебной сети Cisco Learning Network (CLN) по адресу `http://learningnetwork.cisco.com` и присоединяйтесь к группам по изучению CCENT и CCNA. Это позволит участвовать в обсуждениях тем, связанных

с экзаменами CCENT (ICND1) и CCNA (ICND1 + ICND2). Зарегистрируйтесь, присоединитесь к группе и установите фильтр на электронную почту, чтобы перенаправлять сообщения в отдельную папку. Даже если нет времени читать все сообщения сразу, то это можно сделать и позже, когда оно будет, или просмотреть темы сообщений в поисках интересных. Либо можно просто искать интересные сообщения на веб-сайте CLN.

Найдите и распечатайте копию приложения H, "Таблицы для запоминания материала". Это задание используется в большинстве обзоров глав. Поскольку даны незаполненные таблицы из приложения, их заполнение поможет запомнить ключевые факты.

Если вы купили электронную версию книги, найдите и загрузите файлы соответствующих ресурсов (видео и программное обеспечение Sim Lite) согласно инструкции на последней странице файла электронной книги в разделе "Где сопутствующие файлы?"

Установите экзаменационное программное обеспечение PCPT и активизируйте его экзамены. Более подробная информация о загрузке программного обеспечения приведена в разделе "Install the Pearson IT Certification Practice Test Engine and Questions" инструкции.

И наконец, установите программное обеспечение Sim Lit (если еще не куплена полная версия эмулятора). Эмулятор Sim Lit, поставляемый с этой книгой, содержит лишь часть упражнений и лабораторных работ полной версии Pearson Network Simulator.

Итак, приступим

Теперь приступим к первой из многих коротких задач: чтению главы 1. Наслаждайтесь!

Первая часть книги содержит введение в важнейшие темы работы с сетями TCP/IP. В главе 1 представлены термины, концепции и протоколы стека TCP/IP. Главы 2 и 3 рассматривают передачу данных между сетевыми устройствами по физическому каналу связи. Глава 2 посвящена каналам связи между соседними устройствами (локальным сетям), а глава 3 — каналам связи между дистанционными устройствами (распределенным сетям).

В главе 4 речь пойдет о правилах маршрутизации IP, объединяющей каналы связи LAN и WAN при передаче данных от одного пользовательского устройства к другому. И наконец, в главе 5 рассматривается несколько других тем, главным образом связанных с использованием сети TCP/IP приложениями.

Часть I. Основы сетей

Глава 1. “Сетевые модели TCP/IP и OSI”

Глава 2. “Основы сетей LAN”

Глава 3. “Основы сетей WAN”

Глава 4. “Основы IPv4-адресации и маршрутизации”

Глава 5. “Основы протокола TCP/IP: передача данных и приложения”

Обзор части I

Глава 1

Сетевые модели TCP/IP и OSI

Итак, перед вами первая глава учебника по экзаменам CCENT и CCNA! Эта глава начинает часть I, которая посвящена основам работы с сетями. Поскольку сеть требует от всех устройств соблюдения правил, эта часть начинается с обсуждения сетевых моделей, дающего общее представление о сетевых правилах.

Сетевую модель можно представить себе как набор архитектурных планов для строительства дома. Обычно над постройкой дома работает много людей — стекольщики, электрики, каменщики, маляры и др. Но чтобы все столь разные части дома составили единое целое, нужен общий план. Точно так же и люди, создающие сетевые продукты, и люди, которые их используют для создания собственных компьютерных сетей, должны следовать единой сетевой модели. Эта сетевая модель определяет правила работы каждой части сети и их взаимодействия, чтобы вся сеть функционировала правильно.

В экзамене CCNA требуются знания, относящиеся преимущественно к одной модели — TCP/IP (Transmission Control Protocol/Internet Protocol — протокол передачи данных/протокол Интернета). Эта модель активно использовалась на протяжении всей истории развития сетевых технологий, поэтому ее реализацию можно найти практически в каждой существующей на сегодняшний день операционной системе, например, как в мобильных телефонах, так и в высокоуровневых мейнфреймах. Все сети, где встречается оборудование компании Cisco, поддерживают протоколы TCP/IP, поэтому не удивительно, что основное внимание в экзамене уделяется именно этой модели.

В экзамене ICND1, и чуть больше в экзамене ICND2, кроме этого встречаются вопросы по второй распространенной эталонной модели — OSI (Open System Interconnection — модель взаимодействия открытых систем). С исторической точки зрения эталонная модель OSI представляет собой первую попытку создать сетевую модель. Поскольку такая модель была первой и довольно всеобъемлющей, множество терминов в сетевых технологиях взяты из нее или основаны на ее концепциях. Поэтому в данном разделе обсуждаются темы и терминология, связанные с моделью OSI.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

- Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы.
- Выбор компонентов сети, удовлетворяющих заданной спецификации.
- Описание предназначения и основных принципов протоколов в моделях OSI и TCP/IP.

Основные темы

Эта глава знакомит с фундаментальными концепциями компьютерных сетей, а также со структурой двух сетевых моделей: TCP/IP и OSI. Глава начинается с краткого обзора того, как большинство людей представляют себе сеть, что, как мы полагаем, соответствует начальным знаниям до подготовки к экзамену CCNA. Затем будут описаны некоторые из основных характеристик модели TCP/IP. И завершается глава дополнительными концепциями и терминологией, связанной с моделью OSI.

Что такое современные сети

Допустим, вы новичок в компьютерных сетях. Ваши представления о телекоммуникационных сетях, как и у большинства людей, основаны на опыте использования сетевых технологий в качестве пользователя, а не на опыте инженера, который создает компьютерные сети. Например, ваши знания, скорее всего, основаны на опыте использования домашнего подключения к Интернету, возможно, даже вполне высокоскоростного. Возможно, вы пользуетесь компьютером на работе или в учебном заведении для решения каких-либо задач, и такой компьютер обычно подключен к сети с помощью кабеля DSL или телевизионного кабеля (рис. 1.1).

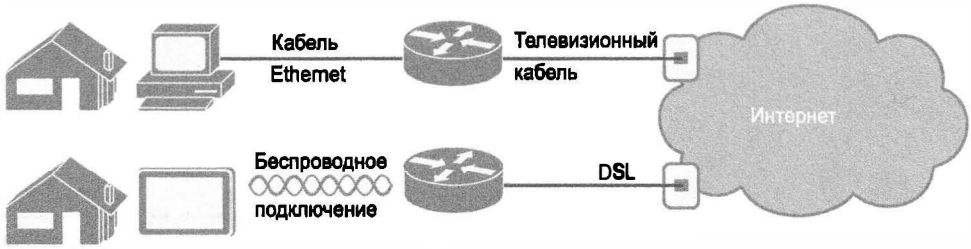


Рис. 1.1. Высокоскоростное соединение с Интернетом с точки зрения конечного пользователя

На рис. 1.1, *сверху*, показан стандартный способ высокоскоростного кабельного подключения к Интернету. К персональному компьютеру пользователя стандартным кабелем Ethernet подключен кабельный модем. В свою очередь модем подключен к розетке кабельного телевидения (Community Antenna Television — CATV) с помощью коаксиального кабеля; точно такой же кабель используется для подключения телевизора. Поскольку такое кабельное подключение к Интернету работает постоянно, пользователю достаточно включить свой компьютер, и он сразу может отправлять электронную почту, искать информацию на веб-сайтах, осуществлять телефонные звонки через сеть и использовать любые другие сетевые приложения.

В нижней части рисунка показано использование двух разных технологий. Сначала вместо кабеля Ethernet планшетный компьютер использует беспроводную технологию, называемую *беспроводной локальной сетью* (wireless LAN, или Wi-Fi). В данном случае для подключения к Интернету маршрутизатор использует другую технологию — *цифровой абонентский канал* (Digital Subscriber Line — DSL).

Экзамены CCNA, в частности экзамен ICND1 (100-101), сосредоточены на технологиях, используемых и для создания домашних сетей (см. рис. 1.1), но в большей

степени они предназначены для корпоративных сетей. Мир информационных технологий (Information Technology — IT) называет сети, созданные некой корпорацией или предприятием для организации взаимосвязи своих сотрудников, *корпоративной сетью* (enterprise network). Когда меньшие домашние сети используются в целях бизнеса, их зачастую называют *сетью малого или домашнего офиса* (Small Office Home Office — SOHO).

У пользователей корпоративных сетей есть некоторое представление о сети своей компании или школы. Люди понимают, что они используют сеть для многих задач. Пользователи персональных компьютеров знают, что их компьютеры подключены кабелем Ethernet к разъему устройства, подключенного к сетевой розетке на стене, как показано на рис. 1.2, *сверху*. Но эти же пользователи могут использовать и беспроводные локальные сети со своим портативным компьютером, когда собираются на совещание в зале заседаний. Обе эти точки зрения конечного пользователя корпоративной сети представлены на рис. 1.2.

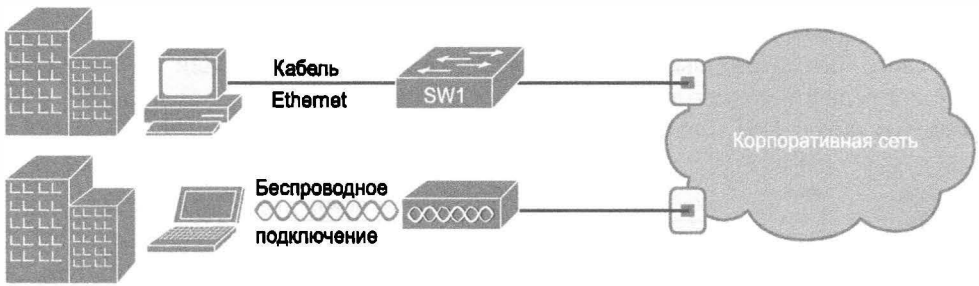


Рис. 1.2. Пример корпоративной сети

ВНИМАНИЕ!

На схемах компьютерных сетей облаком обозначают ту часть сети, детали которой не важны. В рассмотренном выше примере на рис. 1.2 не акцентируется внимание на том, как именно построена или работает корпоративная сеть.

Некоторые пользователи могут вообще не иметь понятия о сети. Вместо этого они просто наслаждаются работой в сети: публикуют сообщения в социальных сетях, обращаются по телефону, ищут информацию в Интернете, слушают музыку и загружают бесчисленное количество приложений на телефоны, не заботясь о том, как все это работает и как их любимое устройство подключено к сети.

Независимо от того, сколько вы уже знаете о работе сети, эта книга и последующая сертификация помогут узнать о работе сети много нового. Работа эта проста: переместить данные с одного устройства на другое. Остальная часть этой главы и первой части книги посвящена основам построения сетей SOHO и корпоративных сетей, способных передавать данные между двумя устройствами.

В деле создания сети обычно довольно много работы осуществляется до того, как будет передан первый пакет. Процесс начинается с планирования: прежде чем строить дом, всегда необходимо знать, как его строить, и составить архитектурные чертежи. Точно так же подход к построению любой компьютерной сети начинается не с монтажа устройств и кабелей, а с изучения архитектурных планов создаваемой сети — модели TCP/IP.

Эталонная модель TCP/IP

Сетевая модель (networking model), называемая также *сетевой архитектурой* (networking architecture), *сетевой схемой* (networking blueprint) или *эталонной моделью*, — это исчерпывающий набор документов. По отдельности каждый документ описывает одну небольшую функцию сети; совместно эти документы определяют все, что необходимо для работы компьютерной сети. Некоторые документы описывают *протокол* (protocol), представляющий собой набор логических правил, которые должны соблюдать коммуникационные устройства. Другие документы определяют некоторые физические требования к сетям. Например, документ может определять уровни напряжения тока, используемые в специфическом кабеле при передаче данных.

Сетевые модели можно считать архитектурными чертежами при строительстве дома. Конечно, дом можно построить и без чертежей. Но чертеж может гарантировать, что дом имеет правильную конструкцию и структуру, поэтому он не завалится и будет иметь соответствующие скрытые пространства для размещения трубопроводов, электрических, газовых магистралей и т.д. Кроме того, благодаря использованию документации множество разных людей, участвуюших в строительстве дома (арматурщики, электрики, каменщики, маляры и т.д.), знают, что, следуя чертежам при выполнении своей части работы, они не создадут проблем для других рабочих.

Точно так же вы можете построить и собственную сеть — написать собственное программное обеспечение, смастерить собственные сетевые карты и все остальное, чтобы получить рабочую сеть. Однако намного проще купить и использовать готовые продукты, которые уже соответствуют некой стандартной сетевой модели или схеме. Поскольку производители сетевых продуктов создавали свои товары с учетом некой сетевой модели, совместно их изделия должны работать хорошо.

История возникновения сетевой модели TCP/IP

Сегодня в мире компьютерных сетей используется только одна сетевая модель: TCP/IP (Transmission Control Protocol/Internet Protocol — протокол управления передачей/протокол Интернета). Но мир не всегда был настолько прост. Когда-то, давным-давно, не было никаких сетевых протоколов, в том числе и протокола TCP/IP. Производители создавали первые сетевые протоколы, но эти протоколы поддерживали только компьютеры данного производителя. Например, корпорация IBM выпустила в 1974 году свою сетевую модель *системной сетевой архитектуры* (Systems Network Architecture — SNA). Другие производители также создавали собственные сетевые модели. В результате, когда компания покупала компьютеры от трех производителей, сетевым инженерам зачастую приходилось создавать три разных сети на базе сетевых моделей, разработанных каждым из производителей, а затем, тем или иным способом, соединять эти сети, получая намного более сложные комбинированные сети. На рис. 1.3, *слева*, приведено общее представление того, как могла выглядеть корпоративная сеть компании в 1980-х годах, до того, как модель TCP/IP получила распространение в объединенных корпоративных сетях.

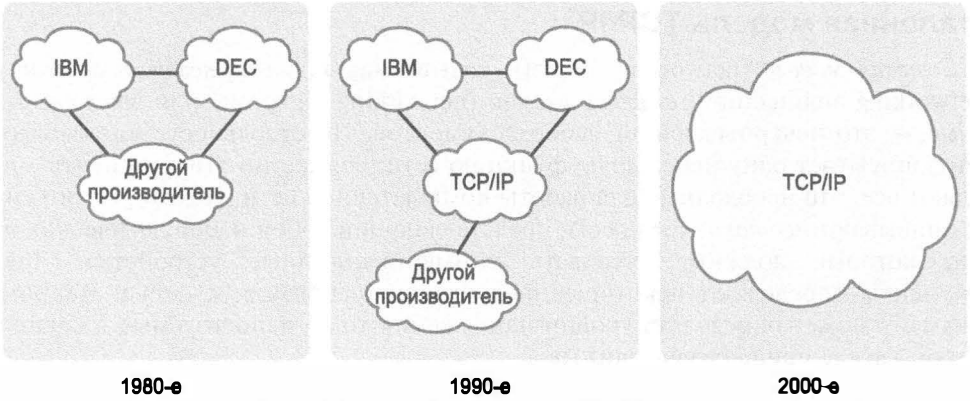


Рис. 1.3. Исторический прогресс: от собственных моделей к открытой модели TCP/IP

Хотя собственные сетевые модели производителей обычно работали хорошо, наличие открытой, независимой от производителя сетевой модели обеспечило бы равенство в конкуренции и уменьшило бы сложность. *Международная организация по стандартизации* (International Organization for Standardization — ISO) взяла на себя эту тяжелую ношу — разработку универсальной модели в конце 1970-х. Таким образом, в начале 80-х появилась сетевая модель, которая известна как эталонная модель *взаимодействия открытых систем* (Open System Interconnection — OSI). Организация ISO совершила благородный поступок и взяла на себя весь труд по созданию модели ISO: упорядочение и стандартизацию существующих на тот момент протоколов и коммуникаций, разработку теоретических основ методов взаимодействия компьютерных систем во всем мире и т.п. Этой благородной цели было посвящено немало времени, кроме того, большинство технологически высокоразвитых стран принимало участие в процессе разработки и стандартизации модели.

Вторая менее формализованная попытка создать открытую, независимую от производителя сетевую модель была предпринята Министерством обороны США в рамках одного оборонного проекта. Множество исследователей, ученых и просто энтузиастов из различных университетов в США принимали участие в разработке и дальнейшем усовершенствовании оригинальной сетевой структуры, которая появилась благодаря Министерству обороны. Попытка создания открытой сетевой модели в конце концов увенчалась успешным набором протоколов, который сегодня известен под названием *стек TCP/IP*.

На протяжении 1990-х годов компании начали внедрять в свои корпоративные сети эталонные модели OSI, TCP/IP или обе вместе. Однако к концу 1990-х модель TCP/IP стала общепринятой, а модель OSI — нет. На рис. 1.3, *посредине*, приведено общее представление корпоративных сетей в это десятилетие, когда сети полагались на несколько сетевых моделей, включая и TCP/IP.

Ныне, в XXI столетии, доминирует модель TCP/IP. Собственные сетевые модели все еще существуют, но от них, по большей части, отказываются в пользу модели TCP/IP. Модель OSI из-за более медленного и чуть более сложного процесса стандартизации, по сравнению с TCP/IP, так никогда и не получила большой популярности на рынке телекоммуникаций. Модель TCP/IP практически полностью (как

она сама, так и составляющие протоколы) первоначально была разработана добровольцами и энтузиастами со всего мира, поэтому она содержит больше протоколов и технологий, чем какая-либо из существовавших и существующих на сегодняшний день моделей (см. рис. 1.3, *справа*).

В этой главе будут описаны некоторые базовые принципы стандартной модели стека TCP/IP. Некоторые интересные факты, связанные со стеком TCP/IP, могут пригодиться в практической работе, тем не менее основная цель изложенного ниже материала — помочь читателю разобраться в том, что такое сетевая модель, сетевая структура и как в действительности они работают.

В этой главе также рассмотрены некоторые из наиболее распространенных терминов модели OSI. Видели ли вы когда-либо или работали за компьютером под управлением полного стека протоколов модели OSI, а не протоколов стека TCP/IP? Вероятно, нет, поскольку она очень мало распространена. Тем не менее с терминологией OSI приходится сталкиваться каждый день. В экзамене ICND1 (Interconnecting Cisco Network Devices 1 — Объединение устройств компании Cisco, часть 1) встречаются вопросы по основам модели OSI, поэтому можно сказать, что данная глава также поможет читателю подготовиться к соответствующему экзамену.

Обзор модели сети TCP/IP

Модель TCP/IP описывает множество протоколов, позволяющих взаимодействовать компьютерам. Подробное описание протоколов, входящих в стандартный набор TCP/IP, представлено в документах, которые называются *запросами на комментарии* (Requests for Comments — RFC). (Вы можете найти документы RFC, используя любой сетевой поисковый механизм.) Модель TCP/IP не дублирует работу, уже проделанную другими организациями по стандартизации или консорциумом производителей, просто ссылаясь на соответствующий стандарт или протокол, созданный этими группами. Например, *Институт инженеров по электротехнике и электронике* (Institute of Electrical and Electronic Engineers — IEEE) определяет локальные сети Ethernet; поэтому модель TCP/IP не определяет сети Ethernet в своих запросах на комментарии, а ссылается на документы IEEE Ethernet.

Компьютер, использующий протоколы TCP/IP, можно сравнить с обычным телефоном. Можно пойти в магазин, торгующий бытовой техникой, и купить телефонный аппарат какой угодно модели и производителя. Тем не менее, если принести его домой и включить в телефонную розетку тем же самым кабелем, каким был подключен старый аппарат, новый телефон будет работать. Производители телефонов знают стандарты телефонии для своей страны и производят телефоны в соответствии с ними.

Аналогично, когда вы покупаете новый компьютер, он, по сути, уже реализует модель TCP/IP, чтобы вы могли взять соответствующие кабели и подключить компьютер к сети. Теперь вы можете использовать веб-браузер для просмотра своего любимого веб-сайта. Почему? Операционная система на компьютере реализует части модели TCP/IP. Встроенная в компьютер плата Ethernet или плата беспроводной сети реализует некоторые стандарты LAN, используемые моделью TCP/IP. Короче говоря, производители, которые создали аппаратные средства и программное обеспечение, реализовали модель TCP/IP.

Чтобы упростить изучение сетевых моделей, каждая из них разделена на несколько функциональных разделов, называемых *уровнями* (layer). Каждый уровень включает протоколы и стандарты, относящиеся к данному функциональному разделу. Фактически есть две альтернативные модели TCP/IP, как показано на рис. 1.4.

Рис. 1.4. Две сетевые модели TCP/IP

Слева представлена первоначальная модель TCP/IP, описанная в документе RFC 1122. Она подразумевает четыре уровня. Два верхних уровня сосредоточены больше на приложениях, которые должны передавать и получить данные. Нижние уровни сосредоточены на передаче битов по каналу связи, где уровень Интернета осуществляет передачу данных по всему пути от отправляющего компьютера на компьютер конечного получателя.

Справа представлена используемая в настоящее время общепринятая модель, дополнительные уровни которой сформированы за счет разделения канального уровня первоначальной модели на два отдельных уровня: канала связи и физического (подобно двум нижним уровням модели OSI). Обратите внимание: модель справа сейчас используется чаще.

ВНИМАНИЕ!

В исходной модели TCP/IP *канальный уровень* (link) называется также *уровнем доступа к сети* (network access) и *уровнем сетевого интерфейса* (network interface).

Большинство из вас уже слышали о некоторых протоколах TCP/IP, таких как перечисленные в табл. 1.1. Более подробная информация о большинстве протоколов и стандартов, перечисленных в этой таблице, приведена далее в книге. Ниже уровни модели TCP/IP рассматриваются подробнее.

Таблица 1.1. Структурная модель TCP/IP и примеры протоколов

Уровень модели TCP/IP	Примеры протоколов
Приложений	HTTP, POP3, SMTP
Транспортный	TCP, UDP
Интернета	IP
Доступа к сети	Ethernet, Point-to-Point Protocol (PPP), T/1

Уровень приложений TCP/IP

Уровень приложений стека TCP/IP предоставляет службы приложениям и программному обеспечению, работающему на компьютере. Сам он не определяет требования непосредственно к приложениям, а стандартизирует службы, которые могут понадобиться приложениям. Например, протокол уровня приложений HTTP (Hypertext Transfer Protocol — протокол передачи гипертекста) определяет, как веб-браузер может запрашивать содержимое веб-страницы с веб-сервера. Другими словами, уровень приложений представляет собой интерфейс между программным обеспечением компьютера и сетью.

Вероятно, наиболее популярным приложением TCP/IP на сегодняшний день является веб-браузер. Многие компании уже поменяли или как раз меняют свое программное обеспечение таким образом, чтобы с ним можно было работать через веб-браузер. К счастью, работать с браузером исключительно просто — нужно всего лишь запустить его на компьютере, потом набрать адрес веб-сайта в строке ввода адреса, и в окне программы появится ожидаемая веб-страница.

Краткий обзор протокола HTTP

Что же в действительности происходит, когда веб-страница появляется в окне браузера?

Предположим, Боб запустил на своем компьютере веб-браузер. Браузер настроен таким образом, что он сразу обращается к стандартной странице веб-сервера его друга Ларри, или, другими словами, к его *домашней странице*. Схема работы браузера и сервера приведена на рис. 1.5.

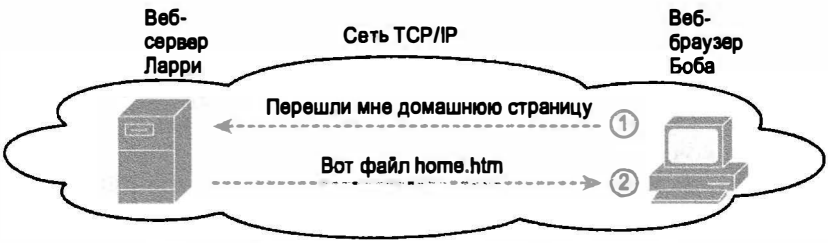


Рис. 1.5. Схема работы приложения с веб-сервером

Итак, что же происходит? Первоначальный запрос от программного обеспечения компьютера Боба запрашивает сервер Ларри об отправке домашней страницы браузеру Боба. Веб-сервер Ларри настроен таким образом, что страница `home.html` является стандартной и в ней содержится домашняя страница Ларри. Программное обеспечение компьютера Боба получает файл страницы от сервера Ларри, и браузер корректно отображает его в своем окне.

Этот пример демонстрирует, как приложения конечной точки (а именно приложение веб-браузера и приложение веб-сервера) используют уровень приложений протокола TCP/IP. Чтобы запросить веб-страницу и возвратить ее содержимое, приложения используют *протокол передачи гипертекста* (Hypertext Transfer Protocol — HTTP).

Протокол HTTP появился в начале 1990-х годов, когда Тим Бернерс-Ли (Tim Berners-Lee) создал первый веб-браузер и веб-сервер. Бернерс-Ли придал протоколу

HTTP возможность запрашивать содержимое веб-страниц, а именно способность веб-браузера запрашивать файлы у сервера, и предоставил серверу возможность возвращать содержимое этих файлов. Общая логика соответствует тому, что изображено на рис. 1.5; а рис. 1.6 демонстрирует ту же идею, но с подробностями, специфическими для протокола HTTP.

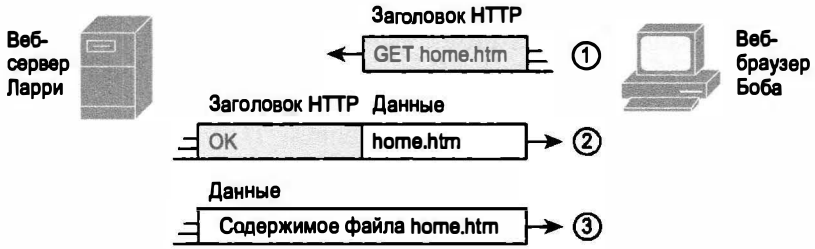


Рис. 1.6. Запрос HTTP GET, ответ HTTP и сообщение с данными

ВНИМАНИЕ!

Полная версия большинства веб-адресов, называемых также *универсальными локаторами ресурсов* (Universal Resource Locators — URL), начинается с символов `http`, что означает использование протокола HTTP для передачи веб-страницы.

Чтобы получить веб-страницу от сервера Ларри (этап 1), Боб пересылает сообщение с заголовком HTTP. Как правило, протоколы используют заголовки как место для размещения используемой ими служебной информации. Этот заголовок HTTP содержит запрос `get` (получить) на получение нужного файла. Обычно такой запрос содержит также имя файла (в данном случае `home.html`), а если имя файла отсутствует, то веб-сервер предполагает, что запрашивается стандартная веб-страница.

Этап 2 на рис. 1.6 демонстрирует ответ веб-сервера Ларри. Сообщение начинается с заголовка HTTP с кодом возврата (200), означающим нечто столь же простое, как "OK". Протокол HTTP определяет также другие коды возврата, таким образом, сервер может указать браузеру, сработал ли запрос. (Еще один пример: при обращении к веб-странице, которая была не найдена, получаешь ошибку HTTP 404 "not found", т.е. передается код возврата HTTP 404.) Второе сообщение включает также первую часть затребованного файла.

Ответ от сервера Ларри также содержит инструкцию HTTP, в заголовке которой написано что-то вроде "OK". Ответ всегда содержит код в заголовке, который указывает запрашивающей стороне, может ли быть выполнен запрос. Например, если серверу приходит запрос на страницу, которая не существует, браузер получит сообщение HTTP с кодом ошибки 404, "страница не найдена". Если же запрашиваемый файл был найден, то в ответ сервер передаст сообщение с кодом 200, который свидетельствует о том, что все в порядке, и выполняется дальнейшая обработка запроса.

Этап 3 на рис. 1.6 демонстрирует еще одно сообщение веб-сервера Ларри веб-браузеру Боба, но на сей раз без заголовка HTTP. Данные по протоколу HTTP передаются при посылке нескольких сообщений, каждое с частью файла. Чтобы не тратить впустую пространство при повторной посылке заголовков HTTP, содержащих ту же информацию, эти последующие сообщения просто опускают заголовок.

Транспортный уровень TCP/IP

Хотя в модели TCP/IP существует множество протоколов уровня приложений, его транспортный уровень содержит меньше протоколов. Двумя наиболее популярными протоколами транспортного уровня модели TCP/IP являются *протокол управления передачей* (Transmission Control Protocol — TCP) и *протокол пользовательских дейтаграмм* (User Datagram Protocol — UDP).

Протоколы транспортного уровня предоставляют службы протоколам уровня приложений, которые в модели TCP/IP располагаются на один уровень выше. Как протокол транспортного уровня предоставляет службы протоколу более высокого уровня? В этом разделе рассматриваются общие концепции на примере одной из служб, предоставляемой протоколом TCP: восстановление при ошибках. В последующих главах транспортный уровень исследуется более подробно и рассматривается куда больше его функций.

Основы восстановления при ошибках протокола TCP

Чтобы представить себе, что именно делает транспортный уровень, нужно сначала обратиться к верхнему уровню, а именно к уровню приложений. Зачем? Каждый уровень многоуровневой модели предоставляет некоторые службы вышестоящему уровню, — так протокол TCP уровня приложений предоставляет службу восстановления при ошибках.

Например, как показано на рис. 1.5, Боб и Ларри используют протокол HTTP для пересылки веб-страницы с веб-сервера на веб-браузер. А что произойдет, если, например, запрос на получение страницы от Боба потеряется где-то по пути? Или ответ от сервера Ларри, содержащий текст веб-страницы, не будет получен? В любом из указанных случаев информация не появится в браузере Боба.

Итак, стеку TCP/IP нужен механизм гарантированной доставки данных в компьютерной сети. Поскольку многим приложениям потребуется такая возможность, создатели протокола TCP включили в него возможность восстановления при ошибках. Для восстановления после ошибок протокол TCP использует концепцию подтверждений. На рис. 1.7 показана основная идея того, как протокол TCP замечает потерю данных и запрашивает у отправителя их повторную передачу.

Рис. 1.7. Служба восстановления при ошибках протокола TCP, предоставляемая протоколу HTTP

Как показано на рис. 1.7, веб-сервер Ларри посылает веб-страницу на веб-браузер Боба, используя три отдельных сообщения. Обратите внимание: здесь пред-

ставлены те же заголовки HTTP, что и на рис. 1.6, а также заголовок TCP. Заголовок TCP демонстрирует порядковый номер (SEQ) каждого сообщения. На этом примере показано: в сети есть некая проблема, не позволившая доставить сообщение TCP (называемое сегментом) с порядковым номером 2. Когда Боб получает сообщения с порядковыми номерами 1 и 3, но не получает сообщения с порядковым номером 2, он понимает, что сообщение 2 было потеряно. Согласно этой логике реализации протокола TCP, он запрашивает у сервера Ларри повторную передачу сообщения 2.

Взаимодействие на смежных и равноправных уровнях

Пример на рис. 1.6 демонстрирует также функцию *взаимодействия на смежных уровнях* (adjacent-layer interaction), которая описывает концепцию взаимодействия смежных уровней сетевой модели на том же компьютере. В этом примере протокол более высокого уровня (HTTP) ожидает восстановления после ошибки. Он просит об этом протокол следующего, нижнего уровня (TCP), и протокол нижнего уровня предоставляет службу уровню выше него.

Рис. 1.6 демонстрирует также пример подобной функции *взаимодействия на равноправных уровнях* (same-layer interaction). Когда определенные слои одного компьютера хотят взаимодействовать с равноправным уровнем на другом компьютере, для хранения информации, которой они хотят обмениваться, эти два компьютера используют заголовки. Например, на рис. 1.6 компьютер Ларри установил порядковые номера 1, 2 и 3, чтобы компьютер Боба мог заметить отсутствие некоторых данных. Процесс передачи на компьютере Ларри создал заголовки TCP с последовательными номерами; процесс на компьютере Боба получает и реагирует на сегменты TCP. Этот процесс, в ходе которого два общающихся по сети компьютера передают и интерпретируют информацию в заголовке, используемом тем же уровнем, называется *взаимодействием на равноправном уровне*.

В табл. 1.2 кратко описаны ключевые моменты взаимодействия смежных уровней на том же компьютере и механизм взаимодействия равноправных уровней на разных компьютерах в сети.

Ключевая тема **Таблица 1.2. Взаимодействие на смежных и равноправных уровнях**

Концепция	Описание
Взаимодействие на равноправных уровнях между разными компьютерами	Два компьютера используют для взаимодействия протокол (и согласованный свод правил). Протоколом называют формальный набор правил, соглашений и форматов, в котором также используются заголовки для упорядоченного обмена информацией между компьютерами. Информация заголовка, добавленная уровнем передающего компьютера, обрабатывается равноправным уровнем получающего компьютера
Взаимодействие на смежных уровнях в одном компьютере	В одном компьютере один уровень может предоставлять некоторые службы другому компьютеру. Программное или аппаратное обеспечение, в котором реализованы процедуры верхнего уровня, запрашивает нижний уровень о выполнении некоторой функции

Сетевой уровень модели TCP/IP

Уровень приложений включает множество протоколов, а транспортный уровень — всего два: TCP и UDP. Основным протоколом сетевого уровня модели TCP/IP является *протокол Интернета* (Internet Protocol — IP). Фактически название *TCP/IP* — это просто названия двух наиболее распространенных протоколов (TCP и IP), разделенные косой чертой.

Протокол IP предоставляет несколько средств, наиболее важными из которых являются адресация и маршрутизация. Этот раздел начинается со сравнения адресации и маршрутизации протокола IP с другой общеизвестной системой почтовой службы, которая использует адресацию и маршрутизацию. Далее в этом разделе содержится введение в IP-адресацию и маршрутизацию. (Более подробная информация по этой теме приведена в главе 4.)

Протокол Интернета и почтовая служба

Предположим, вы написали два письма: одно другу на другой стороне страны и одно другу на другой стороне города. Вы написали адреса на конвертах, наклеили марки и подготовили оба письма к отправке по почте. Есть ли разница в том, как вы готовили каждое письмо? Никакой. Обычно вы просто бросаете их в тот же почтовый ящик, и ожидаете, что почтовая служба доставит оба письма.

Однако почтовая служба должна позаботиться о каждом письме индивидуально и принять решение о том, куда послать каждое письмо, чтобы оно дошло до адресата. Письмо, посланное в пределах города, сотрудникам почтового отделения достаточно поместить в соответствующий грузовик.

Письмо, которое должно пересечь всю страну, почта посылает другому почтовому отделению, которое пересылает его следующему и так далее, пока оно не будет доставлено через всю страну. В каждом почтовом отделении сотрудники должны обработать письмо и решить, куда его послать далее.

Чтобы все это работало, у почтовой службы есть регулярные маршруты для маленьких и больших грузовиков, самолетов, судов, по которым перевозятся письма между почтовыми отделениями. Служба способна получать и передавать письма, при этом она должна принимать правильное решение о том, куда именно послать каждое письмо далее (рис. 1.8).

Рассмотрим в контексте почтовой службы разницу между человеком, посылающим письмо, и сотрудником почты. Отправитель письма ожидает, что почтовая служба доставит письмо куда нужно, однако он не обязан знать точный путь следования письма. Сотрудник почтовой службы, напротив, не пишет письмо, а принимает его от клиента. Но чтобы иметь возможность доставить письмо, он должен быть подробно осведомлен об адресах и почтовых кодах, группирующих адреса в большие группы.

Уровни приложений и транспортные уровни модели TCP/IP выступают в роли человека, посылающего письма через почтовую службу. Эти верхние уровни работают точно так же, независимо от того, находятся ли компьютеры назначения в той же локальной сети или их разделяет Интернет. Посылая сообщение, эти верхние уровни полагаются на уровень, расположенный ниже их, т.е. на сетевой уровень, который должен доставить сообщение.

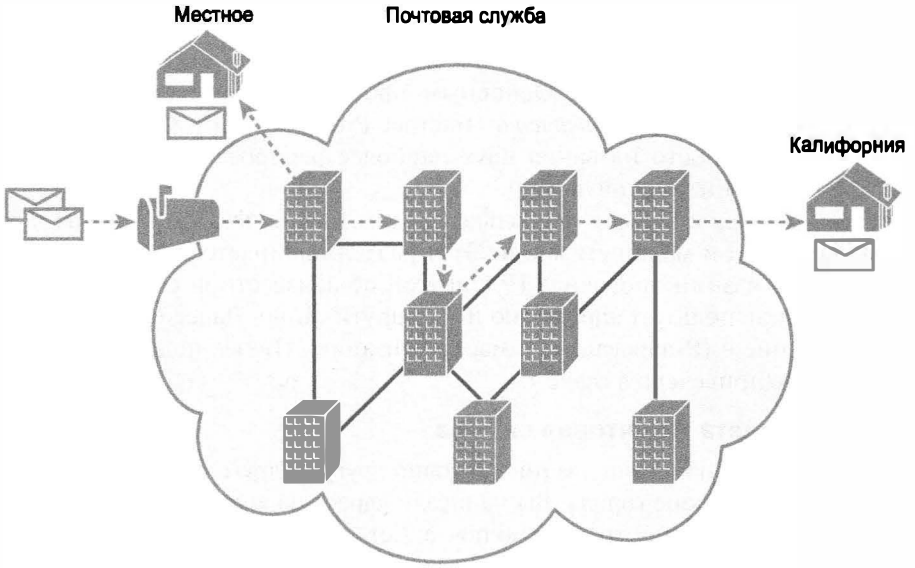


Рис. 1.8. Почтовая служба доставляет письма

Нижние уровни модели TCP/IP выступают в роли почтовой службы, которая правильно доставит сообщения соответствующим получателям. Для этого нижние уровни должны понимать основы физической сети, поскольку они должны выбрать, как лучше доставить данные с одного хоста на другой.

"Так какое же отношение имеет почта к сетевым технологиям?" — спросит читатель? *Протокол Интернета* (Internet Protocol — IP), протокол сетевого уровня модели TCP/IP, работает по тому же принципу, что и почта. Протокол IP определяет адреса для каждого компьютера или хоста в сети, причем каждый хост должен иметь собственный уникальный IP-адрес, точно так же, как и в обычной почте у каждого получателя должен быть свой адрес (город, улица, дом, квартира). На сетевом уровне происходит выбор наилучшего маршрута и пересылка пакета, которую выполняют специализированные устройства — маршрутизаторы. Точно так же как в почтовой службе есть специализированная инфраструктура, состоящая из почтовых отделений, сортировочных машин, грузовиков, самолетов и обученного персонала, данный уровень модели определяет, какая именно инфраструктура нужна, как она должна быть построена и как сеть может доставить данные нужным компьютерам в сети.

ВНИМАНИЕ!

Модель TCP/IP определяет две версии протокола IP: версию 4 (IPv4) и версию 6 (IPv6). В мире все еще главным образом используется протокол IPv4, поэтому во вводной части книги речь везде идет о протоколе IP версии 4. Далее, в части VII, обсуждается более новая версия протокола IP.

Основы адресации протокола Интернета

Протокол IP определяет адреса по нескольким важным причинам. В первую очередь потому, что каждому устройству, которое использует модель TCP/IP (*хосту*

(host) TCP/IP), требуется уникальный адрес, чтобы его можно было идентифицировать в сети. Протокол IP определяет также группировку адресов, аналогично группам в почтовом индексе, используемом почтовой службой США.

Чтобы понять основы, рассмотрим рис. 1.9, на котором показаны знакомый веб-сервер Ларри и веб-браузер Боба; но теперь уже без игнорирования сетевой инфраструктуры между ними.

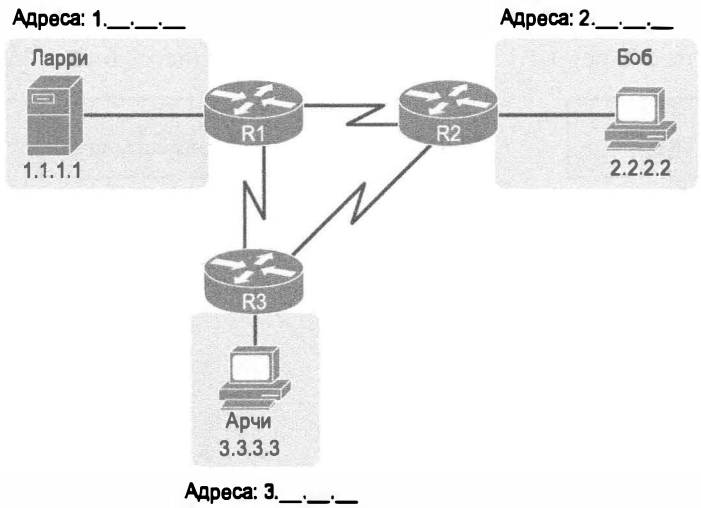


Рис. 1.9. Простая сеть TCP/IP: три маршрутизатора со сгруппированными IP-адресами

В первую очередь обратите внимание на примеры IP-адресов. Каждый IP-адрес содержит четыре числа, разделенных точками. В данном случае для Ларри использован IP-адрес 1.1.1.1, а для Боба — 2.2.2.2. Этот стиль чисел называется *десятичным представлением с разделительными точками* (Dotted-Decimal Notation — DDN).

На рис. 1.9 представлены также три группы адресов. В этом примере все IP-адреса, начинающиеся с 1, должны располагаться в области слева вверху, как показано на рисунке, все адреса, начинающиеся с 2, — справа вверху, а начинающиеся с 3 — внизу.

Кроме того, на рис. 1.9 представлены пиктограммы, которые представляют маршрутизаторы IP. *Маршрутизатор* (router) — это сетевое устройство, соединяющее вместе части сети TCP/IP в целях маршрутизации (пересылки) пакетов IP соответствующему получателю. Маршрутизаторы выполняют работу, аналогичную той, которую выполняют сотрудники почтового отделения: они получают пакеты IP на различных физических интерфейсах и на основании IP-адреса, присвоенного пакету, принимают решение об их пересылке некоторому другому сетевому интерфейсу.

Основы маршрутизации протокола Интернета

Сетевой уровень модели TCP/IP использует протокол IP, предоставляющий службы пересылки пакетов IP от одного устройства другому. Любое обладающее IP-адресом устройство может подключиться к сети TCP/IP и передавать пакеты. В этом разделе представлен простой пример маршрутизации IP.

ВНИМАНИЕ!

Термин *хост IP* (IP host) относится к любому устройству, независимо от его размера или мощности, которое имеет IP-адрес и подключено к любой сети TCP/IP.

На рис. 1.10 повторяется знакомый случай, когда веб-сервер Ларри передает часть веб-страницы браузеру Боба, но теперь с подробностями IP. Обратите внимание: внизу слева у сервера Ларри есть знакомые данные приложения, заголовки HTTP и TCP. Кроме того, сообщение теперь содержит также заголовок IP, включающий IP-адрес отправителя (адрес Ларри 1.1.1.1) и IP-адрес получателя (адрес Боба 2.2.2.2).

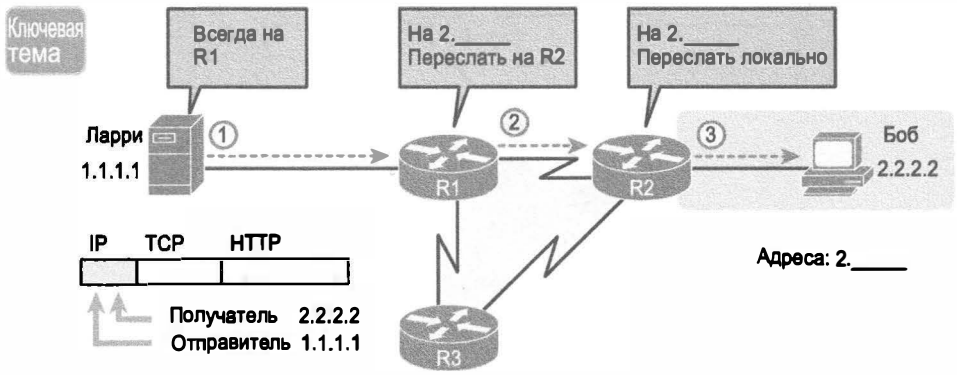


Рис. 1.10. Общая концепция маршрутизации

Первый этап (рис. 1.10, *слева*) начинается с того, что Ларри готов послать пакет IP. Процесс на компьютере Ларри решает послать пакет некоему маршрутизатору (ближайшему маршрутизатору в той же сети LAN), рассчитывая, что он знает, как переслать пакет дальше. (Эта логика очень похожа на нас, когда мы, посылая свои письма, бросаем их в ближайший почтовый ящик.) Ларри не обязан ничего знать ни о топологии, ни о других маршрутизаторах.

На втором этапе маршрутизатор R1 получает пакет IP, и его процесс IP принимает решение. Маршрутизатор R1 исследует адрес получателя (2.2.2.2), сравнивая его с известными ему маршрутами IP, и решает переслать пакет маршрутизатору R2. Этот процесс пересылки пакета IP называется *маршрутизацией IP* (IP routing), или просто *маршрутизацией* (routing).

На третьем этапе маршрутизатор R2 следует той же логике, что и маршрутизатор R1. Его процесс IP сравнивает IP-адрес получателя пакета (2.2.2.2) с известными ему маршрутами IP и решает переслать пакет непосредственно Бобу.

Все экзамены CCNA требуют глубоких знаний протокола IP. Фактически в половине глав этой книги рассматривается некоторое средство, имеющее непосредственное отношение к адресации, маршрутизации IP и тому, как маршрутизаторы осуществляют маршрутизацию.

Канальный уровень TCP/IP (канал связи плюс физический)

Уровень канала связи модели TCP/IP называют еще уровнем соединения хоста и сети. Он стандартизирует аппаратное обеспечение и протоколы, используемые для передачи данных по разным физическим сетям. Термином *канал связи* (link) на-

зывают физические соединения (или каналы) между двумя устройствами и протоколы, используемые для управления этими каналами.

Точно так же как и любой другой уровень в сетевой модели, канальный уровень TCP/IP предоставляет службы вышестоящим уровням. Когда процесс хоста или маршрутизатора IP решает послать пакет IP на другой маршрутизатор или хост, он использует возможности уровня канала связи для передачи этого пакета следующему хосту или маршрутизатору.

Поскольку каждый уровень предоставляет службы уровню выше него, уделим минуту размышлениям о логике IP, связанной с происходящим на рис. 1.10. В этом примере логика IP хоста Ларри принимает решение передать пакет IP на ближайший маршрутизатор (R1), без упоминания о лежащей в основе сети Ethernet. На самом деле для доставки этого пакета с хоста Ларри на маршрутизатор R1 должна использоваться сеть Ethernet, которая реализует протоколы уровня канала связи. На рис. 1.11 демонстрируются четыре этапа процесса, происходящего на уровне канала связи, позволяющего Ларри передать пакет IP маршрутизатору R1.

ВНИМАНИЕ!

На рис. 1.11 сеть Ethernet изображена как серия линий. Сетевые диаграммы зачастую используют это соглашение при изображении локальных сетей Ethernet в случаях, когда фактическая кабельная проводка и устройства LAN не важны для текущего обсуждения, как в данном случае. У реальной сети LAN были бы кабели и такие устройства, как коммутаторы LAN, которые не представлены на этом рисунке.

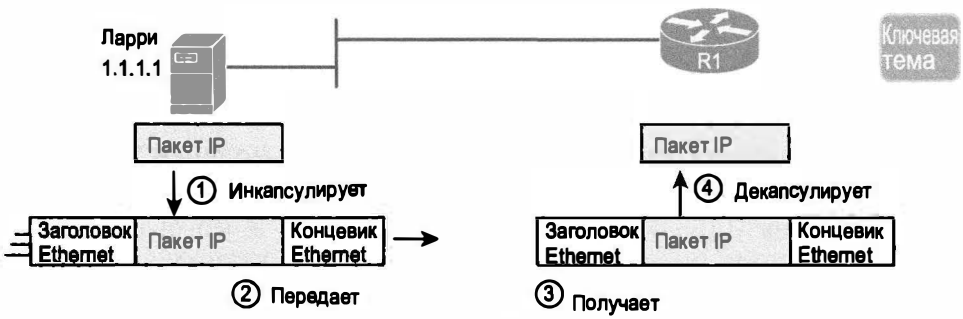


Рис. 1.11. Действия канала связи при передаче пакетов IP между хостами

На рис. 1.11 показаны четыре этапа. Первые два происходят на сервере Ларри, а последние два — на маршрутизаторе R1.

Этап 1 Сервер Ларри инкапсулирует пакет IP между заголовком и концевиком Ethernet, создавая *фрейм* (frame) Ethernet

Этап 2 Сервер Ларри физически передает биты этого фрейма Ethernet, используя электрический ток в кабеле Ethernet

Этап 3 Маршрутизатор R1 физически получает электрический сигнал по кабелю и восстанавливает те же биты, интерпретируя значения электрических сигналов

Этап 4 Маршрутизатор R1 деинкапсулирует пакет IP из фрейма Ethernet, удаляя и отбрасывая заголовок и концевик Ethernet

Таким образом, совместная работа канальных процессов на сервере Ларри и маршрутизаторе R1 позволила доставить пакет с хоста Ларри на маршрутизатор R1.

ВНИМАНИЕ!

Протоколы определяют *заголовки* (header) и *концевики* (trailer) по той же причине, но заголовки располагаются в начале сообщения, а концевики — в конце.

Уровень доступа к сети содержит множество протоколов и стандартов. Например, уровень канала связи включает все варианты протоколов Ethernet наряду с несколькими другими стандартами LAN, которые были популярны в прошедшем десятилетии. Уровень канала связи включает стандарты распределенной сети (Wide-Area Network — WAN) для различных физических сред, которые значительно отличаются от стандартов LAN в связи с большей длиной дистанций, задействованных при передаче данных. Этот уровень включает также популярные стандарты WAN, которые добавляют свои заголовки и концевики, как показано в общем на рис. 1.9, а также такие протоколы, как *протокол двухточечного соединения* (Point-to-Point Protocol — PPP) и Frame Relay. Более подробная информация по этой теме для сетей LAN и WAN приведена в главах 2 и 3 соответственно.

Таким образом, уровень канала связи TCP/IP включает две разные функции: физическая передача данных, а также протоколы и правила, контролирующие использование физической среды. Чтобы соответствовать этой логике, пятиуровневая модель TCP/IP просто разделяет канальный уровень на два уровня (канала связи и физический).

Терминология модели TCP/IP

Прежде чем закончить это введение в модель TCP/IP, рассмотрим некоторые оставшиеся подробности модели и связанную с ней терминологию.

Сравнение первоначальной и модернизированной моделей TCP/IP

Первоначальная модель TCP/IP определяла только один уровень (канала связи) ниже уровня Интернета. Функции, определенные в первоначальном уровне канала связи, могут быть разделены на две основные категории: функции, связанные с физической передачей данных непосредственно, и таковые, связанные с ней косвенно. Например, в четырех этапах, представленных на рис. 1.11, этапы 2 и 3 специфичны именно для передачи данных, а этапы 1 и 4 (инкапсуляции и деинкапсуляция) связаны с ней только косвенно. Это разделение станет понятней по мере изучения дополнительных подробностей каждого протокола и стандарта.

Ныне большинство документов использует более современную версию модели TCP/IP, как показано на рис. 1.12. Их верхние уровни идентичны, только название "Интернет" заменено на "Сетевой". Нижние уровни отличаются. Единый уровень канала связи первоначальной модели разделен на два уровня, чтобы отделить физические подробности передачи от других функций. На рис. 1.12 снова представлены эти две модели, но с акцентом на различия.

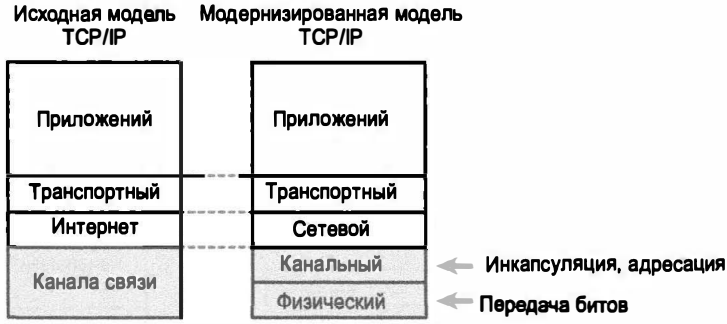


Рис. 1.12. Уровень канала связи по сравнению с канальным и физическим уровнями

Терминология инкапсуляции данных

Выше были рассмотрены принципы работы разных протоколов, HTTP, TCP, IP и Ethernet. Из приведенного описания можно заметить, что каждый из них добавляет собственный заголовок (а для протоколов канала передачи данных еще и концевик) к каждому блоку данных от вышестоящих уровней. *Инкапсуляция* (encapsulation) описывает процесс добавления заголовка (и иногда концевика) к некоторому блоку данных.

Большинство примеров этой главы демонстрируют процесс инкапсуляции. Так, например, протокол HTTP инкапсулирует веб-страницу в заголовок HTTP (см. рис. 1.6). Далее протокол TCP инкапсулирует данные и заголовок протокола HTTP в собственный заголовок (см. рис. 1.7), а протокол IP инкапсулирует все вместе в свой заголовок IP (см. рис. 1.9). В итоге блок данных от уровня Интернета (IP) инкапсулируется в заголовок и концевик протокола канального уровня Ethernet (см. рис. 1.11).

Этап 1 **Создание и инкапсуляция данных уровня приложений в заголовки нужного протокола уровня приложений.** Например, сообщение "`HTTP OK`" может быть помещено в заголовок HTTP и добавлено к блоку данных, содержащему веб-страницу

Этап 2 **Инкапсуляция блока данных от уровня приложений в заголовок транспортного уровня.** Для пользовательских приложений может быть использован протокол TCP или UDP

Этап 3 **Инкапсуляция блока данных от транспортного уровня в заголовок сетевого уровня (т.е. заголовок IP).** Протокол IP определяет IP-адреса, уникально идентифицирующие каждый компьютер

Этап 4 **Инкапсуляция блока данных от сетевого уровня в заголовок и концевик канального уровня.** Этот уровень использует как добавление заголовка, так и концевика

Этап 5 **Передача битов.** На физическом уровне информация кодируется в специальный сигнал, который зависит от среды и технологии передачи фреймов

На рис. 1.13 показана описанная выше концепция; номера слева соответствуют перечисленным этапам передачи информации. Поскольку на уровне приложений далеко не всегда к блоку данных добавляется заголовок, на этом рисунке у уровня приложений отсутствует какой-либо заголовок.

Ключевая тема

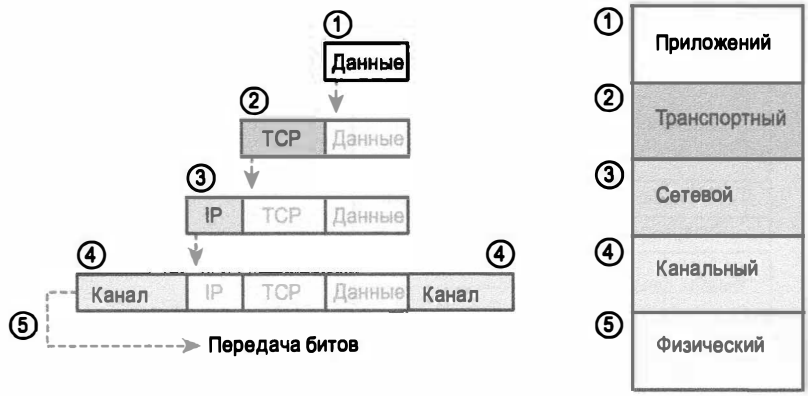


Рис. 1.13. Пять этапов инкапсуляции данных на передающем хосте

Названия сообщений TCP/IP

Следует также обратить пристальное внимание на такие термины, как *сегмент* (segment), *пакет* (packet) и *фрейм* (frame), а также на смысловую нагрузку каждого из них. Каждый из перечисленных терминов описывает инкапсуляцию данных на соответствующем уровне, т.е. добавление заголовка нужного уровня и, возможно, концевика. Каждое из приведенных определений относится к своему собственному уровню: сегмент связан с транспортным уровнем, пакет относится к сетевому уровню, фрейм — к канальному. На рис. 1.14 показаны уровни и соответствующие им блоки данных. Как LH обозначен заголовок канального уровня, а как T — концевик.

Ключевая тема

Рис. 1.14. Смысл терминов "сегмент", "пакет" и "фрейм"

На рис. 1.14 инкапсулированные данные помечены словом "данные". Если сосредоточиться на функциях какого-либо из уровней, то, что находится в поле данных, не представляет для уровня никакого интереса, это просто какой-то блок информации, не имеющий отношения к текущему уровню. Например, в пакете IP может после его заголовка идти заголовок TCP, за ним — заголовок протокола HTTP, дальше будут присутствовать данные какой-либо веб-страницы в поле данных. Однако для протокола IP все, что идет за его собственным заголовком, представляет собой просто некоторые данные. Поэтому на многих схемах, когда иллюстрируют поля пакета IP, все, что идет после заголовка IP, называют данными и не обращают на них ни малейшего внимания.

Эталонная модель OSI

Когда-то многие полагали, что модель OSI выиграет сражение сетевых моделей, обсуждавшихся ранее. Если бы это произошло, то вместо модели TCP/IP на каждом компьютере в мире выполнялась бы модель OSI.

Однако модель OSI не выиграла это сражение. Фактически она больше не существует как сетевая модель, которая применялась бы вместо модели TCP/IP, хотя некоторые из первоначальных протоколов, на которых была основана модель OSI, все еще существуют.

Так почему же модель OSI рассматривается в этой книге? Дело в терминологии. На протяжении тех лет, когда многие были уверены, что модель OSI станет общепринятой во всем мире (главным образом в конце 1980–начале 1990-х), многие производители оборудования и издаваемая документация протоколов начали использовать терминологию из модели OSI. Сегодня эта терминология осталась. Поэтому, хотя работать с компьютером, использующем модель OSI, вероятно, никогда и не придется, чтобы понять современную сетевую терминологию, необходимо узнать кое-что о модели OSI.

Сравнение моделей OSI и TCP/IP

С точки зрения фундаментальных концепций эталонная модель OSI очень похожа на эталонную модель TCP/IP. Она содержит семь уровней, каждый из которых выполняет свои особые функции в сети. Как и уровни модели TCP/IP, каждый из уровней модели OSI ссылается на несколько протоколов и стандартов, которые реализуют функции, определенные каждым уровнем. Для уже существующих протоколов, например стека TCP/IP, новые протоколы и стандарты не разрабатывались, существующие разработки просто были стандартизированы в рамках модели OSI. Например, институт IEEE к тому времени уже выпустил все необходимые спецификации технологии Ethernet, поэтому комитеты OSI не тратили время и ресурсы на то, чтобы выпустить новые стандарты или новый тип технологии Ethernet, они просто ссылались на существовавшие на тот момент стандарты IEEE.

На сегодняшний день модель OSI используется в качестве эталона для сравнения с другими моделями. На рис. 1.15 приведено сравнение семи уровней модели OSI и модели TCP/IP для четырех и пяти уровней.

Ключевая тема

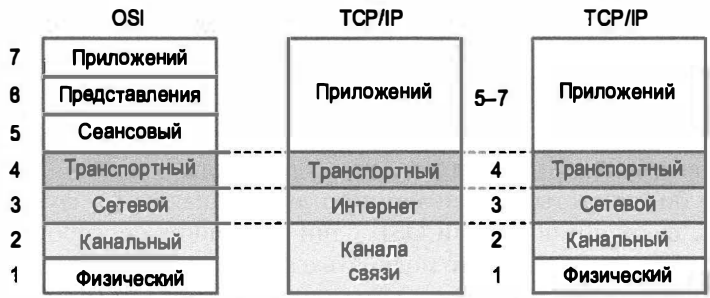


Рис. 1.15. Модель OSI по сравнению с моделью TCP/IP

Далее в этом разделе рассматриваются две области применения терминологии OSI в настоящее время: описание других протоколов и описание процесса инкапсуляции. При этом в тексте кратко описывается каждый уровень модели OSI.

Описание протоколов со ссылками на уровни модели OSI

Даже сейчас сетевые документы зачастую описывают протоколы и стандарты TCP/IP, ссылаясь на уровни модели OSI как по номерам уровней, так и по их названиям. Например, обычное описание коммутатора LAN — "коммутатор второго уровня", где часть "второго уровня" относится к уровню 2 модели OSI. Поскольку у модели OSI был действительно четкий набор функций, связанных с каждым из его семи уровней, зная эти функции, можно легко понять, что подразумевают люди, когда называют продукт или функции по имени уровня модели OSI.

Так, например, уровень Интернета первоначальной модели TCP/IP, единственным протоколом которого является IP, соответствует третьему уровню модели OSI. Поэтому многие специалисты используют терминологию уровней модели OSI и говорят, что протокол IP является протоколом сетевого уровня, или протоколом третьего уровня. В действительности, если использовать нумерацию модели TCP/IP и в общепринятом стиле начинать отсчет уровней снизу, протокол IP относится ко второму или третьему уровню модели в зависимости от используемой версии модели TCP/IP. Но даже при том, что протокол IP относится к протоколу TCP/IP, все используют названия уровней модели OSI и их номера при описании протокола IP и, собственно говоря, любого другого протокола.

Заявления о том, что уровни TCP/IP подобны соответствующим уровням модели OSI, является лишь общим сравнением, а не вполне конкретным. Это сравнение немного напоминает сравнение автомобиля с грузовиком: оба способны доставить из пункта А в пункт Б, но у них есть масса специфических различий: грузовик, например, имеет кузов, специально предназначенный для перевозки груза. Точно так же и сетевые уровни моделей TCP/IP и OSI определяют логическую адресацию и маршрутизацию. Но у их адресов разный размер, и даже логика маршрутизации разная. Таким образом, сравнение уровней OSI с другими моделями — это лишь общее сравнение главных задач, а не сравнение специфических методов.

Функции уровней модели OSI

Компания Cisco требует от специалистов уровня CCNA, чтобы они понимали на базовом уровне функции каждого из семи уровней модели OSI, а также помнили названия всех уровней и порядок их следования. Не менее полезно будет также знать, функциям какого уровня модели OSI наиболее близко соответствует каждое упоминаемое в данной книге устройство или протокол.

Поскольку большинство людей намного лучше знакомы с функциями модели TCP/IP, чем с функциями модели OSI, один из лучших способов узнать о некой функции уровня модели OSI — это подумать о функциях в модели TCP/IP и соотнести их с уровнями модели OSI. Если вы используете модель TCP/IP с пятью уровнями, четыре основных уровня модели OSI практически совпадают с таковыми у модели TCP/IP. Единственное различие в четырех верхних уровнях — это название уровня 3, в модели OSI — это сеть, а в первоначальной модели TCP/IP — Интернет. Три верхних уровня эталонной модели OSI (уровень приложений, представления данных и сеансовый, т.е. уровни 7, 6 и 5) задают те функции, которые относятся к уровню приложений TCP/IP. В табл. 1.3 описаны основные функции всех семи уровней модели OSI.

Таблица 1.3. Функции уровней эталонной модели OSI

Уровень	Описание функций
7	*Уровень приложений* (application layer) является ближайшим к пользователю и предоставляет службы его приложениям. Он является интерфейсом между приложениями и коммуникационным программным обеспечением. Данный уровень также определяет процесс аутентификации пользователя
6	*Уровень представления* (presentation layer) преобразует данные в один из многочисленных существующих форматов, который поддерживается обеими системами и отвечает за согласование формата передачи данных, например, будет это текст в кодировке ASCII[1] или EBCDIC[2], или бинарный файл, или формат BCD[3], или изображение JPEG[4]. Шифрование информации относится к данному уровню и является его службой
5	Как понятно из названия, *сеансовый уровень* (session layer) устанавливает сеансы связи между двумя рабочими станциями, управляет ими и разрывает их. Он также синхронизирует диалог между уровнями представления двух систем и управляет двунаправленным обменом данными так, что приложения верхних уровней уведомляются о получении некоторого завершенного набора сообщений. Этот уровень передает уровню представления данных непрерывный поток данных
4	*Транспортный уровень* (transport layer) сегментирует данные передающей станции и вновь собирает их в единое целое на принимающей стороне. Протоколы этого уровня предоставляют множество служб, которые подробно рассмотрены в главе 6. Уровни 5–7 модели OSI сфокусированы на проблемах приложений, а четвертый уровень связан с проблемами доставки данных дистанционному компьютеру, например, с коррекцией ошибок и контролем потока данных
3	*Сетевой уровень* (network layer) является комплексным уровнем, обеспечивающим выбор маршрута и соединение между собой двух рабочих станций, которые могут быть расположены в географически удаленных друг от друга сетях. Он решает три основные задачи: логическая адресация, маршрутизация (перенаправление пакетов) и определение маршрутов в сети. Концепция маршрутизации определяет, как именно специализированные устройства (обычно это маршрутизаторы) перенаправляют пакеты к конечному получателю. Логическая адресация указывает, как именно должен быть сформирован адрес устройства в сети и как такой адрес будет использоваться в маршрутизации. Механизм определения маршрутов указывает, как именно протоколы маршрутизации способны выяснить абсолютно все маршруты в сети и как выбрать наилучший из них
2	*Канальный уровень* (data link layer) обеспечивает надежную передачу данных по физическому каналу. Он задает правила, определяющие, как именно устройство может переслать данные в определенной среде передачи. Протоколы канального уровня также задают формат заголовков и концевиков второго уровня, которые позволяют успешно передавать данные устройствам в какой-либо среде

[1] Сокращение от American standard code for information interchange — Американский стандартный код обмена информацией. — *Примеч. ред.*

[2] Сокращение от Extended Binary Coded Decimal Interchange Code — расширенный двоично-десятичный код обмена информацией. — *Примеч. ред.*

[3] Сокращение от Binary Coded Decimal — двоично-десятичное число. — *Примеч. ред.*

[4] Сокращение от Joint Photographic Experts Group — объединенная группа экспертов по машинной обработке фотографических изображений, алгоритм сжатия неподвижного изображения. — *Примеч. ред.*

Окончание табл. 1.3

Уровень	Описание функций
1	*Физический уровень* (physical layer) определяет электрические, процедурные и функциональные спецификации для среды передачи данных, в том числе стандартные разъемы, схемы расположения выводов и назначение контактов, уровни напряжений, синхронизацию изменений напряжения, кодирование сигнала в среде, метод модуляции световых сигналов и правила активизации и деактивизации физической среды передачи

В табл. 1.4 перечислены основные устройства и протоколы, наиболее часто встречающиеся в экзамене CCNA и этой книге, а также указана их привязка к уровням модели OSI. Следует заметить, что в действительности большинство из приведенных сетевых устройств работает сразу с несколькими уровнями модели OSI. Указанный в табл. 1.4 уровень является самым верхним, с которым может работать устройство в процессе выполнения своих основных задач. В эталонной модели OSI, если протокол или служба работает на нескольких уровнях, принято указывать самый верхний из них. Например, маршрутизаторы всегда относят к уровню 3 эталонной модели, хотя, вполне очевидно, они содержат функции уровней 1 и 2.

Таблица 1.4. Эталонная модель OSI: примеры устройств и протоколов

Название уровня	Протоколы и спецификации	Устройства
Уровни приложений, представления данных и сеансовый (с 5 по 7)	Telnet, HTTP, FTP, SMTP, POP3, VoIP, SNMP	Хосты, брандмауэры
Транспортный (4)	TCP, UDP	Хосты, брандмауэры
Сетевой (3)	IP	Маршрутизатор
Канальный (2)	Ethernet (IEEE 802.3), HDLC	Коммутатор локальной сети, беспроводная точка доступа, кабельный модем, модем DSL
Физический (1)	RJ-45, Ethernet (IEEE 802.3)	Концентратор LAN, повторитель LAN, кабеля

Кроме того, что на экзамене нужно четко представлять себе основные функции всех уровней модели OSI (см. табл. 1.3) и помнить примеры устройств и протоколов для каждого уровня (см. табл. 1.4), следует также запомнить названия всех уровней. Можно просто вызубрить их, но многие предпочитают использовать некоторые мнемонические правила, чтобы упростить запоминание. Мы предлагаем использовать одну из следуюших ниже фраз, в которых первая буква слова соответствует англоязычному названию соответствующего уровня модели OSI. Уровни в такой схеме запоминания идут в правильном порядке, порядок следования указан в скобках.

- All People Seem To Need Data Processing[5] (слева направо: с 7 по 1).
- Please Do Not Take Sausage Pizzas Away (слева направо: с 1 по 7).
- Pew! Dead Ninja Turtles Smell Particularly Awful (слева направо: с 1 по 7).

[5] В переводе фразы не помогут запомнить названия и порядок следования уровней, следует запоминать их в английском варианте. Первая фраза: *всем людям, несомненно, нужна обработка данных.* Вторая фраза: *пожалуйста, не уносите с собой пиццу с сосисками.* Третья фраза: *Уф! Мертвые черепашки-ниндзя пахнут исключительно ужасно. — Примеч. ред.*

Концепции и преимущества многоуровневой структуры модели OSI

Хотя сетевые модели используют уровни, чтобы классифицировать сетевые функции и помочь людям понять их, для этого есть и другие причины. Рассмотрим, например, еще одну аналогию с почтой. Человек, пишущий письмо, может не думать о том, как почтовая служба доставит письмо через всю страну. Сотрудник почты на полпути следования письма может не задумываться о содержимом письма. Аналогично сетевые модели, которые делят функции на различные уровни, позволяют программным пакетам и аппаратным устройствам реализовать функции определенного уровня и подразумевать, что другое программное обеспечение и аппаратные средства выполнят функции, определенные другими уровнями.

Преимущества многоуровневых сетевых моделей

Ключевая тема

- **Упрощение решаемых задач.** Многоуровневая модель позволяет разделить задачу на меньшие и более простые этапы.
- **Стандартизация интерфейсов** взаимодействия уровней позволяет разным производителям создавать устройства, ориентированные на выполнение какой-либо определенной функции, а конкуренция в рамках открытых моделей ведет к значительному улучшению продукта.
- **Упрощение процесса обучения.** Людям намного проще изучать детали отдельных протоколов и уровней.
- **Упрощение процесса разработки новых устройств.** Чем проще продукты и устройства, тем проще и быстрее можно внести в них какие-либо изменения, а также разработать новые продукты.
- **Совместимость устройств разных производителей.** Если устройства отвечают одним и тем же стандартам, это означает, что компьютеры и сетевое оборудование от разных производителей будет работать корректно.
- **Модульные разработки.** Один производитель может написать программное обеспечение, работающее на верхних уровнях, например веб-браузер компании Opera, а другой разработчик может написать программное обеспечение нижних уровней, например реализацию стека протоколов TCP/IP в операционной системе компании Microsoft, и в рамках стандартов программное обеспечение будет успешно работать с сетью.

Терминология инкапсуляции модели OSI

Подобно модели TCP/IP, модель OSI стандартизирует процесс получения служб верхними уровнями от нижних. Чтобы предоставить службы, каждый уровень использует заголовок, а возможно, и концевик. Нижние уровни модели инкапсулируют данные верхних уровней в заголовок определенного формата. Последний раздел этой главы посвящен терминологии и концепциям инкапсуляции в модели OSI.

В модели TCP/IP используются такие термины, как *сегмент* (segment), *пакет* (packet) и *фрейм* (frame), для описания инкапсулированных данных разных уровней (см. рис. 1.13). В модели OSI используется более общий термин — *блок данных протокола* (Protocol Data Unit — PDU).

Под блоком PDU понимают как биты заголовка и концевика соответствующего уровня, так и сами инкапсулированные данные. Например, пакет IP, который показан на рис. 1.12, согласно терминологии модели OSI, является блоком PDU. Зачастую говорят, что пакет IP является блоком PDU третьего уровня (сокращенно L3PDU), поскольку протокол IP относится к третьему уровню (Layer 3 — L3) модели OSI. Таким образом, вместо терминов *сегмент*, *пакет*, *фрейм* в модели OSI используется обозначение PDU уровня *x* (L*x*PDU), где символом *x* обозначается уровень, обсуждаемый в данный момент.

На рис. 1.16 показан типичный процесс инкапсуляции. Вверху показаны данные уровня приложений и его заголовок, в самом низу — блок L2PDU, который передается уже непосредственно в физический канал.

Ключевая тема

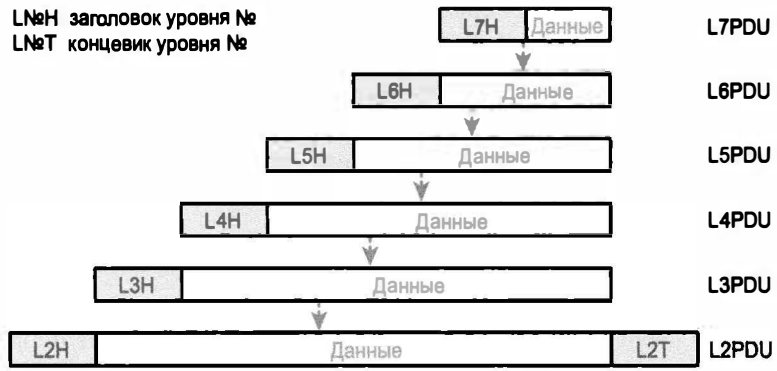


Рис. 1.16. Терминология инкапсуляции

Обзор

Резюме

- Сетевое подключение зависит от способа использования сети; например, домашняя сеть, вероятнее всего, использовала бы подключение DSL или абонентский телевизионный канал, корпоративная сеть — подключение Ethernet или беспроводное подключение.
- *Сетевая модель*, называемая также *сетевой архитектурой* или *схемой сети*, — это исчерпывающий набор документации.
- Современный компьютерной мир использует только одну сетевую модель: протокол TCP/IP.
- Модель TCP/IP определяет и использует множество протоколов, которые и позволяют компьютерам общаться.
- У первоначальной модели TCP/IP было четыре уровня: приложений, транспортный, Интернета и канала связи. Современная модернизированная модель TCP/IP имеет пять уровней: приложений, транспортный и сетевой, а уровень канала связи разделен на два уровня — канальный и физический.
- Протоколы уровня приложений модели TCP/IP обслуживают прикладное программное обеспечение, выполняющееся на компьютере. Уровень приложений определяет не сами приложения, а необходимые им службы.
- На транспортном уровне чаще всего используются два протокола: протокол управления передачей (TCP) и протокол пользовательских дейтаграмм (UDP).
- Протоколы транспортного уровня обслуживают протоколы уровня приложений, располагающиеся в модели TCP/IP уровнем выше.
- Протокол TCP/IP нуждается в механизме, гарантирующем передачу данных по сети. Поскольку гарантия передачи данных через сеть необходима множеству протоколов уровня приложений, создатели включали в протокол TCP средство восстановления при ошибках. Для восстановления после ошибок протокол TCP использует концепцию подтверждений.
- Протокол IP предоставляет множество средств, важнейшими из которых являются адресация и маршрутизация.
- Сетевой уровень модели TCP/IP, используя протокол IP, обеспечивает передачу пакетов IP с одного устройства на другое. Любое устройство с IP-адресом может быть подключено к сети TCP/IP и передавать пакеты.
- Первоначальный уровень канала связи модели TCP/IP определяет протоколы и аппаратные средства, необходимые для доставки данных через физическую сеть. Термин *канал связи* относится к физическим соединениям или каналам связи между двумя устройствами, и к протоколам, контролирующим эти каналы связи.
- Процесс передачи данных хостом TCP/IP может быть разделен на пять этапов. Первые четыре этапа относятся к инкапсуляции, выполняемой четырь-

мя уровнями модели TCP/IP, а последний этап — это фактическая физическая передача данных.

- Модель OSI подразумевает семь уровней: приложений, представления, сеансовый, транспортный, сетевой, канальный и физический.
- Компания Cisco требует, чтобы сертифицированный специалист CCENT имел понятие об основных функциях каждого уровня модели OSI и помнил их названия. Кроме того, для каждого упоминаемого в книге устройства или протокола важно понимать, какие уровни модели OSI ближе всего соответствуют их функциям.
- Поскольку большинство людей намного ближе знакомы с функциями модели TCP/IP, чем с функциями модели OSI, одним из лучших способов изучения функций различных уровней модели OSI является их сопоставление с функциями модели TCP/IP.
- Если используется модель TCP/IP с пятью уровнями, то четыре ее нижних уровня очень похожи на таковые модели OSI. Единственное различие в нижних четырех уровнях — это название третьего уровня: в модели OSI это сетевой уровень, а в первоначальной модели TCP/IP — уровень Интернета.
- Три верхних уровня эталонной модели OSI (приложений, представления и сеансовый (7, 6 и 5)) совместно определяют функции, соответствующие уровню приложений модели TCP/IP.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Какой из перечисленных ниже протоколов относится к транспортному (transport) уровню модели TCP/IP? (Выберите два ответа.)

   А) Ethernet.
   Б) HTTP.
   В) IP.
   Г) UDP.
   Д) SMTP.
   Е) TCP.

2. Какой из перечисленных ниже протоколов относится к канальному уровню модели TCP/IP? (Выберите два ответа.)

   А) Ethernet.
   Б) HTTP.
   В) IP.
   Г) UDP.
   Д) SMTP.
   Е) TCP.
   Ж) PPP.

3. Когда протокол HTTP запрашивает протокол TCP о передаче каких-либо данных и контроле доставки, такой процесс будет примером:

   А) Взаимодействия двух систем на одинаковом уровне.

   Б) Взаимодействия двух смежных уровней.

   В) Эталонной модели OSI.

   Г) Все указанные выше ответы верны.

4. Примером какой технологии является процесс, когда протокол TCP передающего узла маркирует сегмент порядковым номером 1, а принимающий узел отправляет в ответ подтверждение приема с порядковым номером 1?

   А) Инкапсуляция данных.

   Б) Взаимодействие двух систем на одинаковом уровне.

   В) Взаимодействие двух смежных уровней.

   Г) Эталонная модель OSI.

   Д) Все указанные выше ответы верны.

5. Примером какой технологии является процесс, когда служба веб-сервера добавляет к полю данных, в которое помещена веб-страница, заголовок протокола TCP, затем заголовок протокола IP, а потом заголовок и концевик канального уровня?

   А) Инкапсуляция данных.

   Б) Взаимодействие двух систем на одинаковом уровне.

   В) Эталонная модель OSI.

   Г) Все указанные выше ответы верны.

6. Каким из перечисленных ниже терминов называют блок данных, когда он помещен между заголовком и концевиком канального уровня?

   А) Данные.

   Б) Цепочка.

   В) Сегмент.

   Г) Фрейм.

   Д) Пакет.

7. Какой из уровней модели OSI отвечает за логическую адресацию в рамках всей сети и маршрутизацию?

   А) Уровень 1.

   Б) Уровень 2.

   В) Уровень 3.

   Г) Уровень 4.

   Д) Уровень 5, 6 или 7.

8. Какой из уровней модели OSI задает стандарты для кабельной системы и соединений между узлами?

   А) Уровень 1.

   Б) Уровень 2.

   В) Уровень 3.

   Г) Уровень 4.

   Д) Уровень 5, 6 или 7.

9. Какой из перечисленных ниже терминов не является названием уровня в модели OSI? (Выберите два ответа.)

   А) Уровень приложений.

   Б) Канальный уровень.

   В) Уровень передачи.

   Г) Уровень представления.

   Д) Уровень Интернета.

   Е) Сеансовый уровень.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы приведены в табл. 1.5.

Таблица 1.5. Ключевые темы главы 1

Элемент	Описание	Страница
Табл. 1.2	Взаимодействие на смежных и равноправных уровнях	68
Рис. 1.10	Общая концепция маршрутизации	72
Рис. 1.11	Действия канала связи при передаче пакетов IP между хостами	73
Рис. 1.13	Пять этапов инкапсуляции данных на передающем хосте	76
Рис. 1.15	Модель OSI по сравнению с моделью TCP/IP	77
Список	Преимущества многоуровневых сетевых моделей	81
Рис. 1.16	Терминология инкапсуляции	82

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу. В главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

взаимодействие на смежных уровнях (adjacent-layer interaction), деинкапсуляция (decapsulation), инкапсуляция (encapsulation), фрейм (frame), сетевая модель (networking model), пакет (packet), блок данных протокола (Protocol Data Unit — PDU), взаимодействие на равноправном уровне (same-layer interaction), сегмент (segment)

Ответы на контрольные вопросы:

1 Г и Е. **2** А и G. **3** Б. **4** Б. **5** А. **6** Г. **7** В. **8** А. **9** В и Д

Глава 2

Основы сетей LAN

Практически любую корпоративную компьютерную сеть можно разделить по типу технологии на две части: *локальную сеть* (Local-Area Network — LAN) и *распределенную сеть* (Wide-Area Network — WAN). Локальные сети, как правило, объединяют соседние устройства, находящиеся в той же комнате, в том же здании или на той же территории. Глобальные сети, напротив, объединяют устройства, расположенные, как правило, относительно далеко друг от друга. Совместно локальные и глобальные сети формируют единую корпоративную сеть, решающую основную задачу компьютерной сети: передают данные от одного устройства другому.

За прошедшие годы было разработано множество типов локальных сетей, но на сегодняшний день используются два типа: локальные сети Ethernet и беспроводные локальные сети. Локальные сети Ethernet используют для каналов связи между узлами кабели, в которых, как правило, используются медные провода, поэтому локальные сети Ethernet зачастую называют *проводными локальными сетями* (wired LAN). Беспроводные локальные сети не используют ни проводов, ни кабелей, для каналов связи между узлами они используют радиоволны.

Эта глава только знакомит с локальными сетями Ethernet, а их более подробное описание приведено в части II (главы 6–10).

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы.

Выбор компонентов сети, удовлетворяющих заданной спецификации.

Передача данных между двумя хостами по сети.

Выбор подходящей среды, кабелей, портов и разъемов для подключения сетевых устройств Cisco к другим сетевым устройствам и хостам в сети LAN.

Технологии коммутации сетей LAN

Технологии и методы управления доступом к передающей среде для сети Ethernet.

Основные темы

Обзор локальных сетей

Термин *Ethernet* относится к семейству стандартов LAN, совместно определяющих физические и канальные уровни наиболее популярной в мире проводной технологии LAN. Стандарты, выработанные Институтом инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers — IEEE), определяют кабельные соединения, разъемы на концах кабелей, правила для протоколов и все остальное, необходимое для создания локальной сети Ethernet.

Типичные домашние локальные сети

Для начала рассмотрим локальную сеть *малого или домашнего офиса* (Small Office Home Office — SOHO), использующую только технологию Ethernet. В первую очередь сети LAN необходимо такое устройство, как *коммутатор LAN* (LAN switch), предоставляющее несколько физических портов для подключения кабелей. Технология Ethernet использует *кабели Ethernet* (Ethernet cable), к которым относятся любые кабели, соответствующие любому из множества стандартов Ethernet. Кабели Ethernet в локальной сети используются для подключения различных устройств Ethernet и узлов к одному из портов коммутатора Ethernet.

Схема домашней локальной сети приведена на рис. 2.1. На рисунке представлены один коммутатор LAN, пять кабелей и пять других узлов Ethernet: три компьютера, принтер и одно сетевое устройство — *маршрутизатор* (router). (Маршрутизатор соединяет сеть LAN с сетью WAN, в данном случае с Интернетом.)

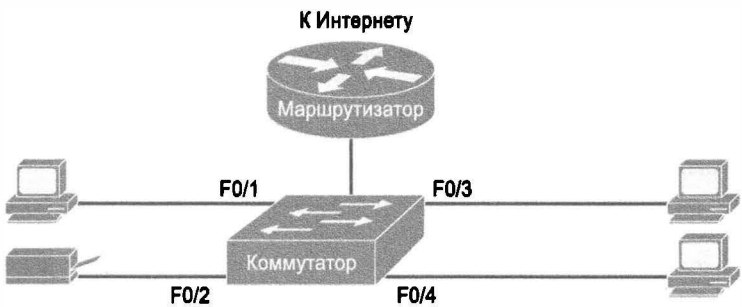


Рис. 2.1. Типичная малая домашняя сеть только на Ethernet

Хотя маршрутизатор и коммутатор на рис. 2.1 — разные устройства, ныне в большинстве домашних локальных сетей Ethernet маршрутизатор и коммутатор объединены в одно устройство. Сейчас продаются интегрированные сетевые устройства потребительского класса, работающие и как маршрутизатор, и как коммутатор Ethernet, а также выполняющие другие функции. Как правило, на упаковке этих устройств написано "маршрутизатор", но большинство моделей имеет также встроенный коммутатор LAN с четырьмя или девятью портами Ethernet.

Типичные современные домашние локальные сети поддерживают также беспроводные соединения LAN. Стандарт Ethernet определяет только проводную техно-

логию LAN; другими словами, локальные сети Ethernet используют только кабели. Однако вполне можно построить единую локальную сеть, использующую одновременно технологию Ethernet и беспроводную технологию, определенную стандартом IEEE. Беспроводные локальные сети, определенные стандартом IEEE начиная с 802.11, используют для передачи битов с одного узла на другой радиоволны.

Большинство беспроводных локальных сетей полагается на еще одно сетевое устройство: беспроводную *точку доступа* (Access Point — AP) LAN. Точка доступа действует наподобие коммутатора Ethernet, позволяя всем беспроводным узлам LAN общаться с коммутатором Ethernet, передавая и получая данные. Конечно, будучи беспроводным устройством, AP не нуждается в кабельных портах Ethernet, кроме одного, необходимого для подключения точки AP к сети LAN Ethernet (рис. 2.2).

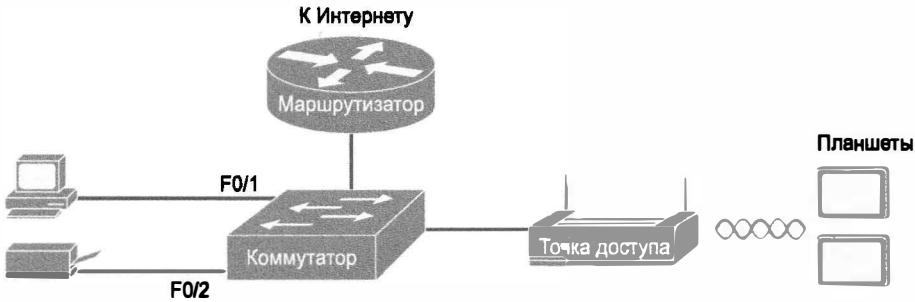


Рис. 2.2. Типичная малая проводная и беспроводная домашняя сеть

Обратите внимание: на этом рисунке маршрутизатор, коммутатор Ethernet и беспроводная точка доступа LAN представлены как три отдельных устройства, чтобы была лучше понятна роль каждого из них. Однако в большинстве современных сетей SOHO использовалось бы единое устройство, обычно называемое "беспроводной маршрутизатор", выполняющее все эти функции.

Типичные корпоративные локальные сети

Корпоративные сети имеют те же потребности, что и сети SOHO, но в много большем масштабе. Например, корпоративная локальная сеть Ethernet начинается с коммутаторов LAN, установленных за закрытыми дверцами в кабельных шахтах на каждом этаже здания. Монтажники протягивают кабели Ethernet от этих кабельных узлов до помещений и залов, где понадобится подключать устройства к сети LAN. Одновременно большинство предприятий поддерживает также беспроводные локальные сети в том же пространстве, позволяя сотрудникам свободно перемещаться, продолжая работать с различными устройствами, не обладающими интерфейсом LAN Ethernet.

Концептуальное представление типичной корпоративной сети LAN в трехэтажном здании приведено на рис. 2.3. На каждом этаже есть коммутатор Ethernet и беспроводная точка доступа. Для связи между этажами коммутатор каждого этажа подключен к единому центральному коммутатору. Например, компьютер PC3 может отправить данные на компьютер PC2, но сначала они направились бы через коммутатор SW3 на первый этаж общему коммутатору SWD, а затем отправились бы назад через коммутатор SW2 на второй этаж.

Ключевая тема

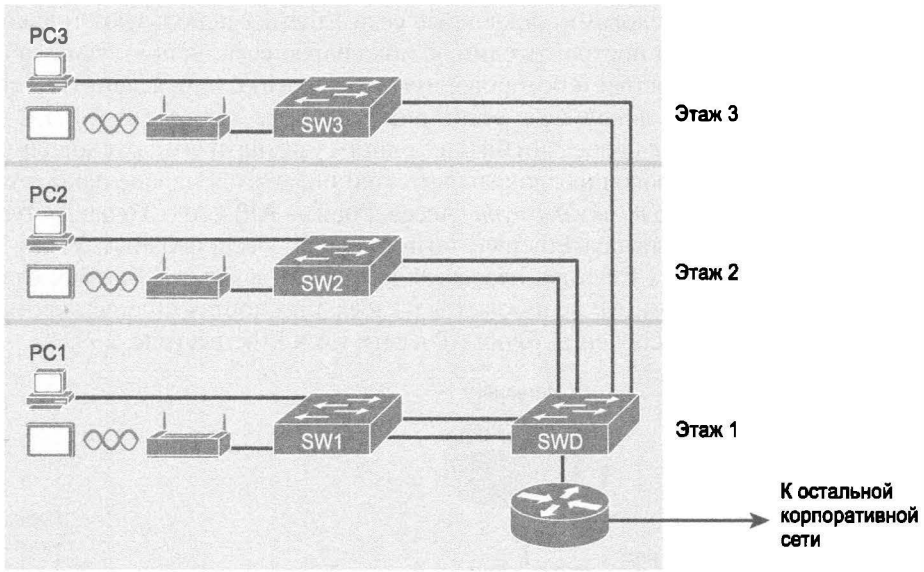


Рис. 2.3. Типичные проводная и беспроводная корпоративные локальные сети

На рисунке демонстрируется также типичный способ подключения сети LAN к сети WAN с использованием маршрутизатора. Саму сеть LAN формируют коммутаторы и беспроводные точки доступа. Маршрутизаторы подключены как к сети LAN, так и WAN. Для подключения к сети LAN маршрутизатор использует интерфейс Ethernet LAN и кабель Ethernet, как показано на рис. 2.3, *внизу справа*.

Оставшаяся часть этой главы посвящена исключительно Ethernet.

Разнообразие стандартов физического уровня Ethernet

Термин *Ethernet* относится ко всему семейству стандартов. Одни стандарты определяют специфические особенности передачи данных по конкретному типу кабельного соединения на конкретной скорости. Другие определяют протоколы, или правила, которым должны следовать узлы Ethernet, чтобы быть частью сети LAN Ethernet. Все они изданы IEEE и содержат число 802.3 в начальной части названия стандарта.

Поскольку технология Ethernet существует приблизительно 40 лет, она охватывает довольно большое разнообразие физических каналов связи. Ныне технология Ethernet поддерживает множество стандартов для различных видов оптических и медных кабелей и скоростей от 10 Мбит/с до 100 Гбит/с. Стандарты определяют также разные типы кабелей и допустимую для них максимальную длину.

Фундаментальным критерием для выбора кабеля является материал его проводов, используемых для физической передачи битов: это либо медь, либо оптическое волокно. *Неэкранированная витая пара* (Unshielded Twisted Pair — UTP) дешевле оптоволоконного кабеля. Для передачи данных между узлами Ethernet такой кабель использует электрические провода. Оптоволоконный кабель дороже, он передает данные в виде светового потока, распространяющегося внутри стеклянного оптиче-

ского волокна в сердцевине кабеля. Хотя волоконно-оптические кабели дороже, они допускают более длинные дистанции между соединяемыми узлами.

Чтобы быть в состоянии выбрать приобретаемые изделия для новой сети LAN Ethernet, сетевой инженер должен как минимум знать названия различных стандартов и средств Ethernet, поддерживаемых данными изделиями. Определяя стандарты физического уровня Ethernet, IEEE использовал несколько соглашений об именовании. Официальное название начинается с числа 802.3, сопровождаемого символьным суффиксом. Для стандарта IEEE используются также и более осмысленные названия, включающие в себя указание скорости и сокращенные сведения о кабеле UTP (с суффиксом "T") или оптоволоконном кабеле (с суффиксом "X").

Список некоторых стандартов физического уровня Ethernet приведен в табл. 2.1. В таблице содержится достаточно много имен, чтобы дать представление о формате соглашений об именовании IEEE. В ней перечислены также четыре наиболее распространенных стандарта, подразумевающих использование кабелей UTP, поскольку обсуждение Ethernet в этой книге сосредоточено главным образом на возможностях канала UTP.

Ключевая тема

Таблица 2.1. Некоторые из типов локальных сетей Ethernet

Скорость	Общеизвестное название	Неофициальное название стандарта IEEE	Официальное название стандарта IEEE	Тип кабеля, максимальная длина (м)
10 Мбит/с	Ethernet	10BASE-T	802.3	Медный, 100
100 Мбит/с	Fast Ethernet	100BASE-T	802.3u	Медный, 100
1000 Мбит/с	Gigabit Ethernet	1000BASE-LX	802.3z	Оптический, 5000
1000 Мбит/с	Gigabit Ethernet	1000BASE-T	802.3ab	Медный, 100
10 Гбит/с	10 Gig Ethernet	10GBASE-T	802.3an	Медный, 100

ВНИМАНИЕ!

Внутри оптоволоконного кабеля находятся длинные тонкие стеклянные волокна. Соединяемые узлы Ethernet передают по стеклянному оптическому волокну свет, кодируя биты изменением его интенсивности.

Канальный уровень Ethernet обеспечивает единообразие поведения всех каналов связи

Хотя технология Ethernet имеет много стандартов физического уровня, она действует как единая технология LAN, поскольку использует единый стандарт канального уровня для всех типов физических каналов связи Ethernet. Этот стандарт определяет единый для всех заголовок и концевик Ethernet. (Напомним, что заголовок и концевик — это дополнительные наборы байтов до и после данных, используемые для осуществления действий по передаче данных в сети LAN.) Независимо от того, передаются ли данные по кабелю UTP или оптоволоконному кабелю, а также независимо от скорости передачи, заголовок и концевик канала связи имеют одинаковый формат.

В то время как стандарты физического уровня сосредоточены на передаче битов по кабелю, протоколы канала связи Ethernet — на передаче *фреймов Ethernet* (Ethernet frame) от узла отправителя к узлу получателя Ethernet. С точки зрения канала связи узлы создают и пересылают фреймы. Как упоминалось в главе 1, сам термин *фрейм* (frame) относится к заголовку и концевику протокола канала связи, заключающим между собой данные. Узлы Ethernet просто перенаправляют фрейм по необходимым каналам связи, чтобы доставить его соответствующему получателю. Пример процесса приведен на рис. 2.4. В данном случае компьютер PC1 посылает фрейм Ethernet компьютеру PC3. Фрейм следует по каналу UTP к коммутатору Ethernet SW1, а затем по каналу оптоволоконного кабеля с коммутатора Ethernet SW2 на коммутатор SW3 и, наконец, по еще одному кабелю UTP на компьютер PC3. Обратите внимание, что биты в этом примере фактически передаются на четырех разных скоростях: 10 Мбит/с, 1 Гбит/с, 10 Гбит/с и 100 Мбит/с соответственно.

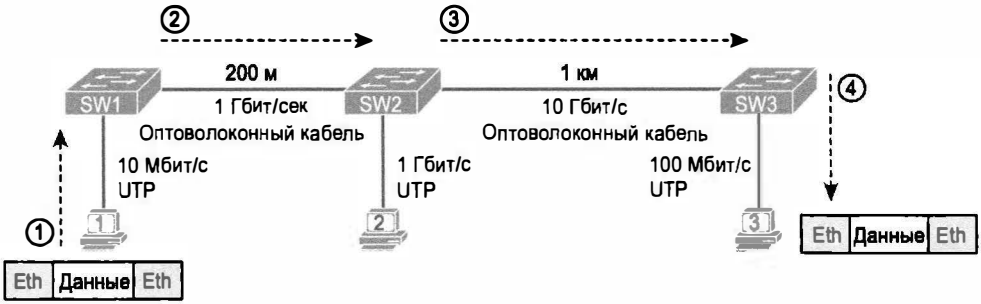


Рис. 2.4. Локальная сеть Ethernet передает фрейм по каналам связи многих типов

Так что же такое локальная сеть Ethernet? Это комбинация пользовательских устройств, коммутаторов LAN и различных видов кабелей. Каждый канал связи может использовать кабели различных типов и различные скорости передачи. Однако они взаимодействуют между собой, чтобы доставить фреймы Ethernet с одного устройства LAN на другое.

В оставшейся части этой главы эти концепции рассматриваются немного глубже: сначала — подробнее о построении физической сети Ethernet, затем обсуждение некоторых правил маршрутизации фреймов по локальной сети Ethernet.

Построение физических сетей Ethernet на базе UTP

В данном разделе речь пойдет об отдельных физических каналах связи между любыми двумя узлами Ethernet. Прежде чем сеть Ethernet сможет передавать фреймы Ethernet между пользовательскими устройствами, каждый узел должен быть готов и способен отправлять данные по конкретному физическому каналу связи. В этом разделе рассматриваются некоторые из частностей передачи данных по каналам связи Ethernet.

Здесь рассматриваются три наиболее часто используемых стандарта Ethernet: 10BASE-T (Ethernet), 100BASE-T (Fast Ethernet, или FE) и 1000BASE-T (Gigabit Ethernet, или GE), а также подробности передачи данных в обоих направлениях по кабелю UTP. И наконец, рассматривается конструкция кабелей UTP, используемых при передаче на скоростях 10, 100 и 1000 Мбит/с.

Передача данных по витой паре

Когда Ethernet передает данные по кабелям UTP, физически данные передает электрический ток, текущий по проводам в кабеле UTP. Чтобы лучше понять, как Ethernet передает данные при помощи электричества, разделим концепцию на две части: как создать электрический канал связи и как заставить электрический сигнал передавать единицы и нули.

Технология Ethernet определяет способ использования двух проводов витой пары для создания одного электрического канала связи, как показано рис. 2.5. Вместо всего кабеля UTP, соединяющего два узла, на рисунке представлены только два его провода. Чтобы электрический ток мог течь, необходима замкнутая цепь, поэтому два узла используют контакты портов Ethernet для подключения проводов пары так, чтобы замкнуть цепи и позволить электричеству течь.

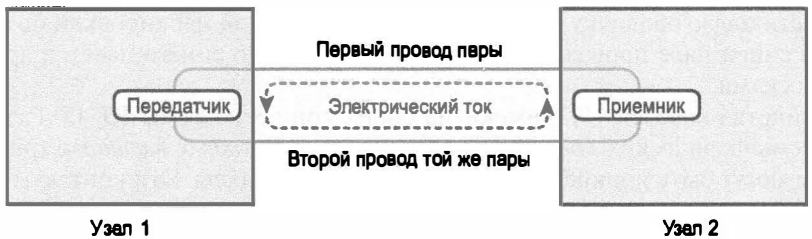


Рис. 2.5. Создание одного электрического канала для передачи в одном направлении по одной паре

Для передачи данных эти два устройства должны выполнять некоторые правила — *схему модуляции* (encoding scheme). Это похоже на то, как два человека разговаривают на одном языке: говорящий произносит слова на некоем языке, а слушатель, поскольку он тоже говорит на том же языке, может понять эти слова. Передающий узел изменяет электрический сигнал согласно схеме модуляции, а принимающий узел, используя те же правила, интерпретирует эти изменения как нули и единицы. (Например, канал 10BASE-T использует схему модуляции, согласно которой двоичному нулю соответствует переход от более высокого напряжения к низкому на протяжении примерно 1/10 000 000 секунды.)

Обратите внимание: в реальном кабеле UTP провода будут скручены вместе, а не идти параллельно, как на рис. 2.5. Скручивание помогает решить некоторые физические проблемы передачи. Когда электрический ток течет по любому проводу, он создает *электромагнитные помехи* (Electromagnetic Interference — EMI), т.е. помехи, накладывающиеся на электрические сигналы в соседних проводах, включая провода в том же кабеле. (Электромагнитные шумы между проводными парами в том же кабеле вызываются *перекрестными помехами* (crosstalk).) Скручивание пар проводников позволяет компенсировать большую часть электромагнитных шумов, поэтому большинство сетевых кабелей, использующих медные провода, содержат витые пары.

Разделение канала связи Ethernet UTP

Термин *канал связи Ethernet* (Ethernet link) относится к любому физическому кабелю между двумя узлами Ethernet. Чтобы узнать, как работает канал связи Ethernet UTP, разделим физический канал связи на составляющие части, как показано на рис. 2.6: сам кабель, разъемы на его концах и соответствующие порты на устройствах, в которые должны быть вставлены разъемы.

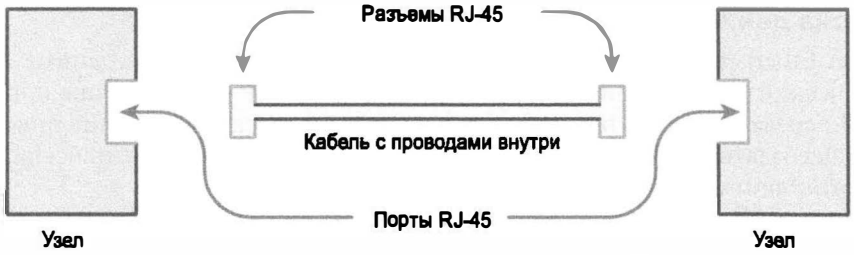


Рис. 2.6. Основные компоненты канала связи Ethernet

Сначала рассмотрим сам кабель UTP. Он содержит несколько медных проводов, попарно скрученных в витые пары. Стандартам 10BASE-T и 100BASE-T достаточно двух пар проводов, а стандарту 1000BASE-T требуется четыре пары. Каждый провод имеет пластиковую оболочку с соответствующей цветовой маркировкой по схеме. Например, в синей паре проводов один провод монотонно синего цвета, а другой с белыми полосками.

Большинство кабелей UTP имеют на обоих концах разъемы RJ-45. Разъем RJ-45 имеет восемь физических *контактов* (pin), или *контактных площадок* (pin position), к которым могут быть прикреплены восемь проводов кабеля. Эти контакты обеспечивают передачу электрического тока между концами медных проводов кабеля и электроникой узла.

ВНИМАНИЕ!

Если это возможно, найдите ближайший кабель UTP и рассмотрите его разъемы. Рассмотрите контактные площадки и цвета проводов в разъеме.

Физический канал связи завершается *портом Ethernet* (Ethernet port) узла или портом RJ-45. Он соответствует разъему RJ-45 на конце кабеля, позволяя подключить его к узлу. У персональных компьютеров порт RJ-45 Ethernet может располагаться на дополнительной *плате сетевого интерфейса* (Network Interface Card — NIC) или может быть встроен непосредственно в систему. У коммутатора, как правило, есть несколько портов RJ-45, поскольку они предоставляют пользовательским устройствам место для подключения к LAN Ethernet.

Фотографии кабелей, разъемов и портов приведены на рис. 2.7.

ВНИМАНИЕ!

Разъем RJ-45 чуть шире, но в остальном похож на разъем RJ-11, который обычно используется для телефонных кабелей.

На рисунке слева представлен разъем RJ-45 с восемью контактными площадками на конце, а порты справа. Вверху справа — сетевая плата Ethernet, еще не установленная в компьютер. Внизу справа представлена обратная сторона коммутатора Cisco 2960 с несколькими портами RJ-45, позволяющими подключить к сети Ethernet несколько устройств.

И наконец, хотя разъемы RJ-45 для кабелей UTP весьма распространены, коммутаторы LAN компании Cisco оснащены также разъемами и других типов. Покупая одну из множества моделей коммутатора Cisco, следует учесть количество и типы необходимых физических портов.

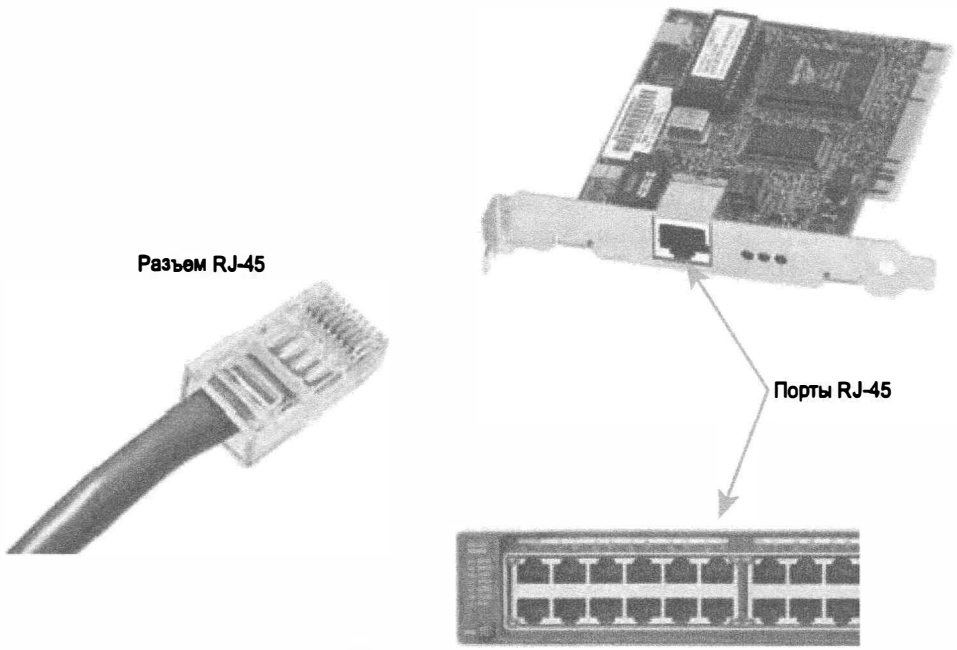


Рис. 2.7. Разъемы и порты RJ-45

Для расширения возможностей по типам кабелей Ethernet, даже после покупки коммутатора Cisco некоторые из его физических портов можно заменить. Один из типов такого порта, *гигабитовый конвертер интерфейса* (Gigabit Interface Converter — GBIC), появился на рынке практически одновременно с Gigabit Ethernet, поэтому ему и было дано такое название. Позже появился улучшенный и уменьшенный сменный интерфейс, *малый сменный форм-фактор* (Small Form-Factor Pluggable — SFP), позволяющий пользователям заменять аппаратные средств и изменять тип физического канала связи. На рис. 2.8 представлена фотография коммутатора Cisco с переходником, вынутым из слота SFP.

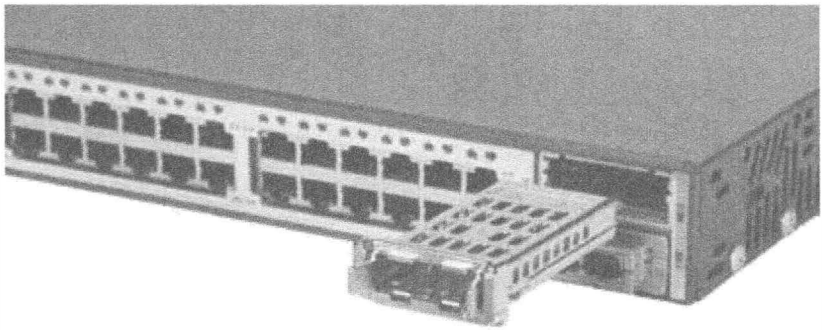


Рис. 2.8. Гигабитовый оптоволоконный порт, вынутый из порта SFP коммутатора

Схема расположения контактов кабелей UTP 10BASE-T и 100BASE-T

Описанное в этом разделе до сих пор напоминало управление грузовиком на 1000-гектарном ранчо при полном игнорировании правил дорожного движения. Работая на собственном ранчо, вполне можно управлять грузовиком на всей территории, в любом месте, и полиция не будет возражать. Но как только вы появитесь на общественной дороге, полицейские потребуют, чтобы вы вели себя согласно правилам. Аналогично в этой главе до сих пор обсуждались лишь общие принципы передачи данных без детализации ряда важнейших правил для кабеля Ethernet (подобных правилам дорожного движения), благодаря которым все устройства отправляют данные по правильным проводам в кабеле.

Следующий подраздел посвящен соглашению для стандартов 10BASE-T и 100BASE-T, поскольку они используют кабель UTP подобным способом (в частности, используются только две пары проводов). Стандарт 1000BASE-T (Gigabit Ethernet) подразумевает использование четырех пар проводов.

Схема расположения выводов прямого кабеля

Стандарты 10BASE-T и 100BASE-T подразумевают использование двух пар проводов кабеля UTP, по одной для каждого направления, как показано на рис. 2.9. На рисунке показаны четыре провода, все из одного кабеля UTP, который соединяет компьютер и коммутатор LAN. В данном случае компьютер слева передает по верхней паре, а коммутатор справа — по нижней.

Ключевая тема

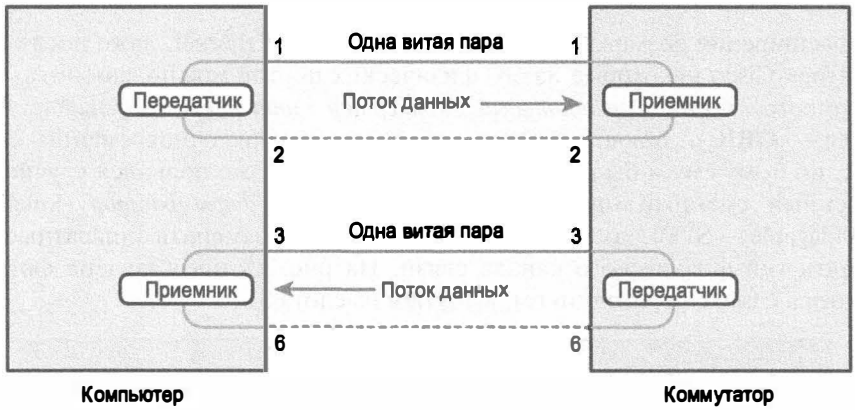


Рис. 2.9. Принцип передачи по более чем двум электрическим каналам (каждый в одном направлении) между двумя узлами Ethernet

Для правильной передачи по каналу связи провода в кабеле UTP должны быть подключены к правильным контактным площадкам разъемов RJ-45. Например, на рис. 2.9 передатчик компьютера слева должен быть подключен к тем контактным площадкам двух проводов, которые следует использовать для передачи. Эти два провода должны быть подключены к правильным контактам разъема RJ-45 на коммутаторе, чтобы плата получателя на коммутаторе могла использовать их.

Чтобы понять схему расположения выводов кабеля (какой провод с каким контактом на концах кабеля должен быть соединен), сначала следует понять, как рабо-

тают сетевые платы и коммутаторы. Как правило, передатчики сетевой платы Ethernet используют пару, подключенную к контактам 1 и 2; получатель сетевой платы использует пару проводов на контактных площадках 3 и 6. С учетом этих фактов порты коммутаторов LAN устроены противоположно: их получатели используют пару проводов на контактах 1 и 2, а их передатчики — проводную пару на контактах 3 и 6.

Чтобы сетевая плата компьютера могла общаться с коммутатором, кабель UTP должен использовать *схему расположения выводов прямого кабеля* (straight-through cable pinout). Термин *схема расположения выводов* (pinout) относится также и к окраске проводов, подключенных к каждой из восьми пронумерованных контактных площадок разъема RJ-45. Прямой кабель Ethernet соединяет провод на контакте 1 одного конца кабеля с контактом 1 на другом конце кабеля; контакт 2 на одном конце с контактом 2 на другом; контакт 3 с контактом 3 и т.д. Кроме того, он использует только провода пары на контактах 1 и 2 и второй пары в контактах 3 и 6.

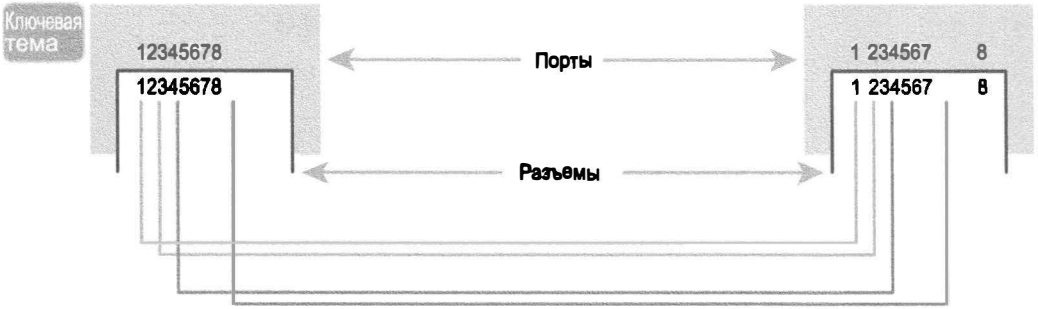


Рис. 2.10. Схемы расположения выводов прямого кабеля Ethernet на 10 и 100 Мбит/с

На рис. 2.11 приведена окончательная схема расположения выводов прямого кабеля. В данном случае компьютер Ларри соединен с коммутатором LAN. На этом рисунке также показан не кабель UTP, а находящиеся внутри него провода, чтобы продемонстрировать концепцию проводных пар и контактов.

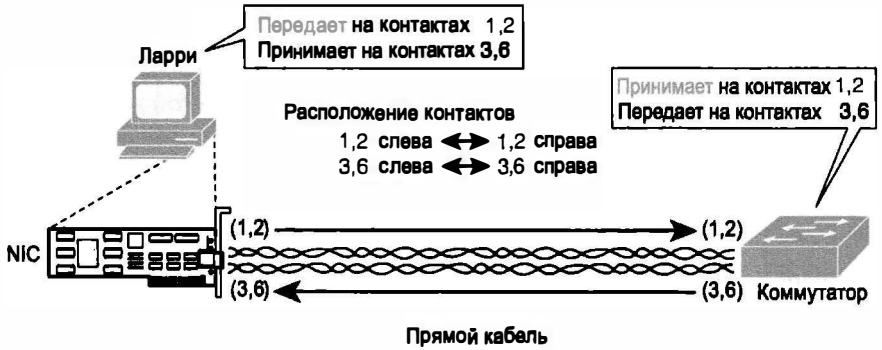


Рис. 2.11. Концепция прямого кабеля Ethernet

Схема расположения выводов перекрещенного кабеля

Прямой кабель работает правильно, когда для передачи данных узлы используют противоположные пары. Но когда два устройства соединены каналом связи Ethernet, оба они передают по тем же контактам. В этом случае необходим кабель с другим типом схемы расположения выводов — *перекрещенный кабель* (crossover cable). Схема расположения выводов перекрещенного кабеля подразумевает соединение передающих контактов на каждом конце с приемными контактами на противоположном конце.

Вышесказанное проще понять на рис. 2.12: на нем демонстрируется происходящее в канале связи между двумя коммутаторами. Оба коммутатора передают данные на паре контактов 3 и 6, а получают на паре контактов 1 и 2. Таким образом, кабель должен соединить пару контактов 3 и 6 на каждой стороне с контактам 1 и 2 на противоположной стороне, чтобы соединить передатчики с приемниками. Вверху на рисунке приведены номера контактов схемы расположения выводов, а внизу их концептуальное представление.

Ключевая тема

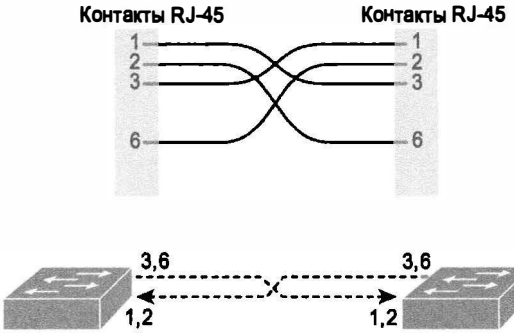


Рис. 2.12. Схемы расположения выводов перекрещенного кабеля Ethernet на 10 и 100 Мбит/с

Выбор правильной схемы расположения выводов кабеля

На экзамене следует уверенно выбирать тип кабеля (прямой или перекрестный), необходимый для каждой части сети. Главное, знать, действует ли устройство как сетевая плата компьютера, передающая на контактах 1 и 2, или как коммутатор, передающий на контактах 3 и 6. Затем достаточно применить следующую логику.

Перекрещенный кабель — если конечные точки передают на той же паре контактов.
Прямой кабель — если конечные точки передают на разных парах контактов.

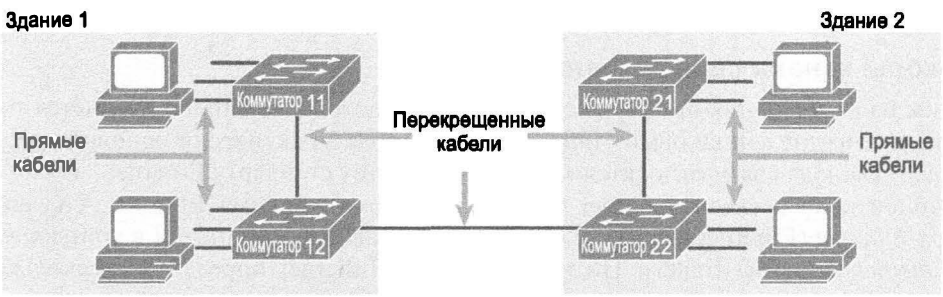
В табл. 2.2 приведен список упоминаемых в этой книге устройств и пар используемых ими контактов (для стандартов 10BASE-T и 100BASE-TX).

Ключевая тема

Таблица 2.2. Устройства, передающие на проводной паре 1,2 и паре 3,6

Передача на контактах 1,2	Передача на контактах 3,6
Сетевые адаптеры персонального компьютера	Концентраторы
Маршрутизаторы	Коммутаторы
Беспроводные точки доступа (интерфейс Ethernet)	—

Пример на рис. 2.13 демонстрирует локальную сеть района и отдельного здания. В данном случае для соединения компьютеров с коммутаторами используется несколько прямых кабелей. Кроме того, для соединения коммутаторов требуются перекрещенные кабели.



Ключевая тема

Рис. 2.13. Пример интерфейсов, использующих полный и полудуплексный режимы

ВНИМАНИЕ!

Если у читателя есть некоторый опыт в создании локальных сетей, может возникнуть мысль, что он где-то использовал неправильный кабель, но сеть все же работала. В коммутаторах компании Cisco реализована функция *автоопределения типа кабеля* (auto-mdix), за счет которой устройство само обнаруживает, какой тип кабеля подключен в порт. Эта функция перенастраивает коммутирующие микросхемы под установленный кабель. Для сдачи экзамена необходимо идентифицировать правильный тип кабеля так, как показано на рис. 2.13.

Схемы расположения выводов кабеля UTP для стандарта 1000BASE-T

Схема расположения выводов кабеля 1000BASE-T (Gigabit Ethernet) отличается от таковой у кабелей 10BASE-T и 100BASE-T. Во-первых, стандарт 1000BASE-T требует четырех пар проводов. Во-вторых, он использует усовершенствованную электронику, позволяющую передавать и получать данные на каждой проводной паре одновременно. Однако схема расположения контактов стандарта 1000BASE-T почти совпадает с прежними стандартами, добавлены лишь подробности для двух дополнительных пар.

Прямой кабель соединяет между собой все восемь контактов с одинаковыми номерами на каждом конце кабеля: контакт 1 с контактом 1, контакт 2 с контактом 2 и так до 8. Пары контактов 1 и 2, а также 3 и 6 остаются, как в прежнем расположении. Добавлена пара контактов 4 и 5, а также заключительная пара контактов 7 и 8 (см. рис. 2.12).

Перекрещенный кабель Gigabit Ethernet перекрещивает те же двухпроводные пары, что и перекрещенный кабель других типов Ethernet (пары на контактах 1,2 с парой на контактах 3,6), а также две новые пары (пары на контактах 4,5 с парой на контактах 7,8).

Передача данных в сетях Ethernet

Хотя стандарты физического уровня изменились незначительно, другие части стандартов Ethernet не зависят от типа физического канала связи Ethernet. В заклю-

чительном разделе этой главы рассматривается несколько протоколов и демонстрируется, что технология Ethernet использует их независимо от типа канала связи. В частности, в этом разделе исследуются детали протокола канального уровня Ethernet, а также то, как узлы, коммутаторы и концентраторы Ethernet передают фреймы Ethernet по локальной сети.

Протоколы канала связи Ethernet

Одним из наибольших преимуществ семейства протоколов Ethernet является то, что они используют одинаковый стандарт канала связи. Фактически основные части стандарта канала связи относятся к первоначальному стандарту Ethernet.

Протокол канала связи Ethernet определяет формат фрейма Ethernet: сначала *заголовок Ethernet* (Ethernet header), в середине передаваемые данные и в конце *концевик Ethernet* (Ethernet trailer). По сути, стандарт Ethernet определяет несколько альтернативных форматов заголовка, но в настоящий момент общепринятым является формат, представленный на рис. 2.14.

Рис. 2.14. Наиболее популярные форматы фреймов Ethernet

В табл. 2.3 приведен список полей заголовка и концевика и их краткое описание. Более подробная информация о некоторых из этих полей приведена ниже.

Таблица 2.3. Поля в заголовке и концевике фрейма стандарта IEEE 802.3

Поле	Длина поля в байтах	Описание или функция
Преамбула (preamble)	7	Синхронизация
Разделитель начала фрейма (Start Frame Delimiter — SFD)	1	Сигнализирует о начале фрейма. Следующий байт — первый байт в адресе получателя
MAC-адрес получателя (destination MAC address)	6	Задает получателя фрейма
MAC-адрес отправителя (source MAC address)	6	Задает отправителя фрейма
Тип (type)	2	Определяет тип протокола, помещенного во фрейм (обычно определяет версию протокола IP: IPv4 или IPv6)
Данные и заполнение (data and pad)*	46-1500	Содержит данные верхних уровней, обычно блок L3PDU (пакет IPv4 или IPv6). Отправитель добавляет заполнение, чтобы это поле удовлетворяло требованию минимальной длины этого поля в 46 байтов
Контрольная сумма фрейма (Frame Check Sequence — FCS)	4	Используется для проверки фрейма на целостность и отсутствие ошибок

*Стандарт IEEE 802.3 ограничивает максимальный размер поля данных величиной в 1500 байтов. Поле данных предназначено для передачи пакета третьего уровня модели OSI; термин *максимальный блок передачи* (Maximum Transmission Unit — MTU) используется для указания максимального размера пакета третьего уровня, который можно переслать через среду передачи данных. Поскольку пакет третьего уровня должен находиться в поле данных фрейма Ethernet, 1500 байт — наибольший размер блока MTU протокола IP, разрешенный в среде Ethernet.

Адресация Ethernet

Поля адресов отправителя и получателя играют важнейшую роль в логике Ethernet, включенной в сертификацию CCENT и CCNA. Их общая идея относительно проста: передающий узел помещает свой адрес в поле адреса отправителя, а адрес устройства Ethernet назначения — в поле адреса получателя. Отправитель передает фрейм, ожидая, что локальная сеть Ethernet доставит его указанному получателю.

Адреса Ethernet, называемые также адресами *контроля доступа к среде передачи* (Media Access Control — MAC), или *MAC-адресами* (MAC address), являются двоичными числами длиной 6 байтов (48 бит). Для удобства на большинстве компьютеров MAC-адреса представлены как шестнадцатеричные числа с 12-ю цифрами. Для удобства чтения MAC-адреса на некоторых устройствах Cisco могут быть разделены также точками; например, коммутатор Cisco мог бы представить MAC-адрес как 0000.0C12.3456.

Большинство MAC-адресов идентифицирует одну сетевую плату или даже один порт Ethernet, поэтому их зачастую называют *одноадресатным адресом* (unicast address) Ethernet. Термин *одноадресатный* является просто формальным способом указать на тот факт, что адрес представляет один интерфейс локальной сети Ethernet. (Этот термин контрастирует также с двумя другими терминами типов адресов Ethernet, *широковещательным* и *многоадресатным*, которые рассматриваются далее в этом разделе).

Идея отправки данных к получателю по одноадресатным MAC-адресам в целом работает хорошо, но только если все одноадресатные MAC-адреса уникальны. Если две сетевые платы попытаются использовать тот же MAC-адрес, возникнет беспорядок. (Проблема подобна беспорядку, вызванному на почте наличием двух одинаковых адресов для двух разных клиентов: почтальон не будет знать, кому именно нужно доставить письмо.) Если два компьютера в той же сети Ethernet попытаются использовать тот же MAC-адрес, возникнет вопрос, на какой из них должны поступать фреймы, посланные на этот MAC-адрес?

Эта проблема решается сугубо административно, всем устройствам Ethernet при изготовлении присваивают всемирно уникальный MAC-адрес. Перед началом выпуска изделий Ethernet изготовитель должен запросить у IEEE всемирно уникальный 3-байтовый код изготовителя — *уникальный идентификатор организации* (Organizationally Unique Identifier — OUI). Изготовитель обязуется присваивать всем своим сетевым платам (и другим изделиям Ethernet) MAC-адреса, начинающиеся с полученного 3-байтового OUI. Значение последних 3 байтов числа изготовитель также обязуется никогда не повторять с этим OUI. В результате MAC-адрес каждого устройства уникален.

ВНИМАНИЕ!

В стандарте IEEE такие MAC-адреса называются также глобальными.

Структура одноадресатного MAC-адреса с идентификатором OUI представлена на рис. 2.15.

Рис. 2.15. Формат MAC-адресов Ethernet

Адреса Ethernet известны под многими названиями: адрес LAN, адрес Ethernet, аппаратный адрес, встроенный адрес, физический адрес, универсальный адрес или MAC-адрес. Например, термин *встроенный адрес* (Burned-In Address — BIA) означает, что это постоянный MAC-адрес, прошитый в микросхеме ROM сетевой платы. Термин *универсальный адрес* (universal address) подчеркивает тот факт, что адрес, присвоенный изготовителем сетевой плате, должен быть уникальным среди всех MAC-адресов в мире.

Кроме одноадресатных адресов, технология Ethernet использует также адреса групп. *Адрес группы* (group address) идентифицирует несколько интерфейсных плат LAN. Посланный на адрес группы фрейм может быть доставлен некоему набору устройств в локальной сети или даже всем устройствам. Фактически IEEE определил две общие категории для адресов групп Ethernet:

- *Широковещательный адрес* (broadcast addresses). Фреймы, посланные по этому адресу, должны быть доставлены всем устройствам сети Ethernet. Имеет значение FFFF.FFFF.FFFF.
- *Многоадресатный адрес* (multicast addresses). Фреймы, посланные по этому адресу, должны быть скопированы и перенаправлены на то подмножество устройств локальной сети, которые добровольно соглашаются получать фреймы, отправленные на определенный многоадресатный адрес.

В табл. 2.4 приведена краткая информация по разновидностям MAC-адресов.

Таблица 2.4. Терминология и функции MAC-адресов в локальной сети

Термин или функция	Описание
MAC-адрес	*Контроль доступа к среде передачи* (Media Access Control). Стандарт IEEE 802.3 (Ethernet) описывает подуровень MAC среды Ethernet
Адрес Ethernet, адрес сетевой платы, адрес локальной сети	Другие названия, часто используемые вместо термина *MAC-адрес*. Шестибайтовый адрес адаптера локальной сети
Прошитый адрес (Burned-in address)	Шестибайтовый адрес, назначенный производителем сетевого адаптера

Окончание табл. 2.4

Термин или функция	Описание
Одноадресатный адрес (Unicast address)	Синоним MAC-адреса, описывающий единственное устройство в локальной сети
Широковещательный адрес (Broadcast address)	Адрес, обозначающий "все устройства в локальной сети в данный момент"
Многоадресатный адрес (Multicast address)	В среде Ethernet групповой адрес идентифицирует некую группу устройств в пределах одной локальной сети

Идентификация протокола сетевого уровня по полю типа Ethernet

В то время как поля адреса заголовка Ethernet играют важную и вполне очевидную роль в локальных сетях Ethernet, поле Type Ethernet играет менее очевидную роль. Поле Type Ethernet, или EtherType, находится в заголовке канального уровня Ethernet и принимает участие в сетевых вычислениях на маршрутизаторах и хостах. В основном поле Type идентифицирует тип сетевого уровня (уровень 3) пакета, находящегося во фрейме Ethernet.

Сначала рассмотрим содержимое части данных фрейма Ethernet, представленного ранее на рис. 2.14. Как правило, он содержит пакет сетевого уровня, созданный в соответствии с протоколом сетевого уровня на некоем устройстве в сети. Исторически такими протоколами были Systems Network Architecture (SNA) от IBM, Novell NetWare, DECnet от Digital Equipment Corporation и AppleTalk от Apple Computer. Ныне наиболее распространенным протоколом сетевого уровня является протокол IP протокола TCP/IP в версии 4 (IPv4) или 6 (IPv6).

Хост отправителя имеет возможность вставить в заголовок значение (шестнадцатеричное число), идентифицирующее тип пакета, инкапсулируемого во фрейме Ethernet. Но какое это должно быть число, чтобы идентифицировать пакет как имеющий тип IPv4? Или тип IPv6? IEEE публикует список значений EtherType, согласно которому у каждого протокола сетевого уровня, нуждающегося в индивидуальном значении EtherType, есть свое число. Отправителю достаточно знать список. (Любой желающий может просмотреть список на сайте `www.ieee.org` и поискать ***EtherType***.)

Например, хост может послать один фрейм Ethernet с пакетом IPv4, а следующий с пакетом IPv6. Как показано на рис. 2.16, у каждого фрейма могло бы быть иное значение поля Type Ethernet, зарезервированное IEEE.

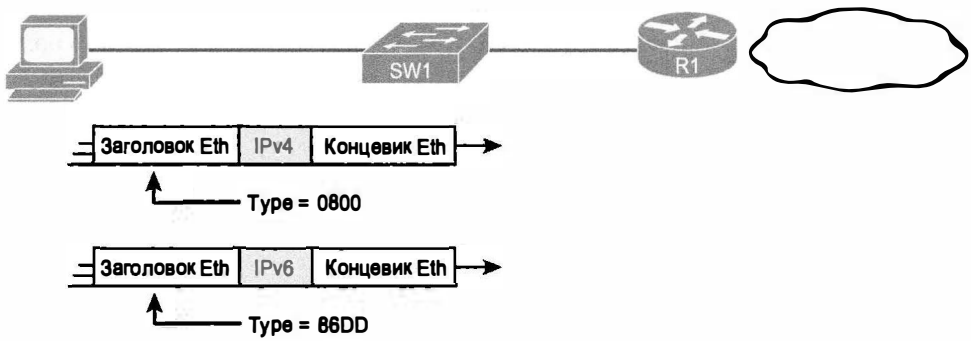


Рис. 2.16. Использование поля типа Ethernet

Обнаружение ошибок при помощи поля FCS

Стандарт Ethernet предоставляет узлам способ выяснить, не изменились ли биты фрейма при пересечении канала связи Ethernet. (Обычно биты изменяются из-за электрических помех или неисправности сетевой платы.) Для обнаружения ошибок протокола канала связи Ethernet, как и большинство других включенных в экзамены CCNA протоколов канала связи, использует поле в концевике фрейма.

Контрольная сумма фрейма (Frame Check Sequence — FCS) Ethernet — единственное поле в концевике Ethernet — позволяет получающему узлу сравнить полученный результат с отправленным и выяснить, не произошло ли ошибки. Перед отправкой фрейма отправитель применяет к фрейму сложную математическую формулу и сохраняет результат в поле FCS. Получатель применяет к полученному фрейму ту же математическую формулу, а затем сравнивает свои результаты с результатами отправителя. Если результаты совпадают, значит, фрейм не изменился; в противном случае произошла ошибка, и получатель отбрасывает фрейм.

Обратите внимание, что *обнаружение ошибок* (error detection) не означает *восстановления после ошибок* (error recovery). Стандарт Ethernet определяет, что фрейм с ошибкой просто отбрасывается без попыток его восстановления. Другие протоколы, особенно TCP, восстанавливают потерянные данные. Заметив потерю, они запрашивают повторную передачу данных.

Передача фреймов Ethernet коммутаторами и концентраторами

Локальные сети Ethernet ведут себя немного по-разному в зависимости от наличия более современных устройств, в частности коммутаторов LAN, или более старых устройств, а именно концентраторов LAN. Более современные коммутаторы позволяют использовать дуплексную логику передачи, которая намного быстрее и проще полудуплексной логики, присущей концентраторам. Этому фундаментальному различию посвящена заключительная тема данной главы.

Передача в современных локальных сетях Ethernet в дуплексном режиме

Современные локальные сети Ethernet используют множество физических стандартов Ethernet, но со стандартными фреймами Ethernet, передаваемыми по физическим каналам связи любых поддерживаемых типов. Каждый канал связи может работать на разных скоростях, но каждый канал связи позволяет подключенным узлам пересылать биты фрейма следующему узлу. Чтобы доставить данные от передающего узла Ethernet к узлу назначения, они должны взаимодействовать.

Процесс относительно прост, что позволяет каждому устройству посылать больше фреймов за секунду. На рис. 2.17 приведен пример передачи фрейма Ethernet с компьютера PC1 на компьютер PC2.

Ниже описаны этапы, показанные на рисунке.

1. Компьютер PC1 создает и отправляет исходный фрейм Ethernet, используя как адрес отправителя собственный MAC-адрес и MAC-адрес компьютера PC2 как адрес получателя.
2. Коммутатор SW1 получает фрейм Ethernet и передает его через свой интерфейс G0/1 (сокращение от Gigabit interface 0/1) на коммутатор SW2.

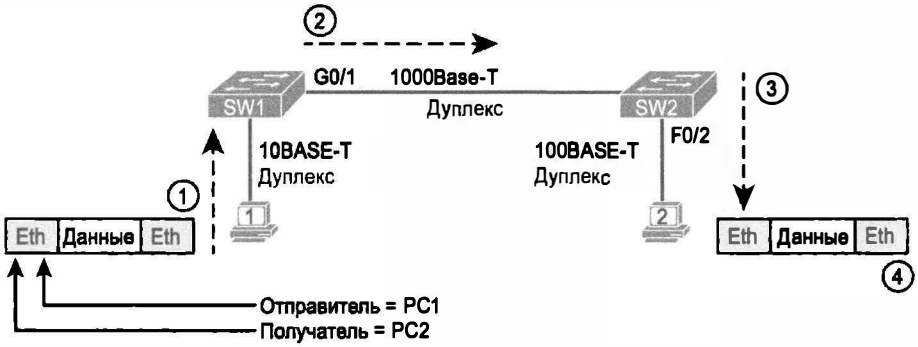


Рис. 2.17. Пример передачи данных в современной сети Ethernet

3. Коммутатор SW2 получает фрейм Ethernet и передает его через интерфейс F0/2 (сокращение от Fast Ethernet interface 0/2) на компьютер PC2.
4. Компьютер PC2 получает фрейм, распознает MAC-адрес получателя, а поскольку это его собственный адрес, обрабатывает фрейм.

Сеть Ethernet на рис. 2.17 использует дуплексный режим на каждом канале связи, но распознать это не всегда просто. Дуплексный режим означает, что ни сетевая плата, ни порт коммутатора не имеет полудуплексных ограничений. Рассмотрим, что такое дуплексный и полудуплексный режимы.

Определения полудуплексного и дуплексного режимов

Ключевая тема

Полудуплексный режим (half-duplex). Логика передачи, при которой порт отправляет данные, только когда он не получает данные. Другими словами, нельзя одновременно передавать и получать данные.

Дуплексный режим (full-duplex). Отсутствие полудуплексного ограничения.

Поэтому при наличии лишь компьютеров и коммутаторов LAN и отсутствии концентраторов LAN все узлы могут использовать дуплексный режим. Все узлы могут передавать и получать данные через порт одновременно. Например, на рис. 2.17 компьютеры PC1 и PC2 могут одновременно передавать фреймы друг другу без всяких полудуплексных ограничений.

Использование полудуплексного режима концентраторами LAN

Чтобы понять логику полудуплексного режима, имеет смысл узнать немного больше об уже устаревшем сетевом устройстве — концентраторе LAN. Когда в 1990 году IEEE впервые ввел стандарт 10BASE-T, коммутаторов LAN он еще не включал. Концентратор LAN, подобно коммутатору LAN, предоставлял несколько портов RJ-45 для подключения каналов связи с компьютерами, но он использовал другие правила перенаправления данных.

Для перенаправления данных концентраторы LAN используют стандарты физического уровня, т.е. это устройство уровня 1. Когда электрический сигнал поступает на один порт концентратора, он повторяет его на всех других портах (кроме порта, на который поступает сигнал). В результате данные поступают на все подключенные к концентратору узлы в надежде, что они достигнут получателя. У концентратора нет таких концепций, как фреймы Ethernet, адреса и т.д.

Недостаток использования концентраторов LAN в том, что если два устройства или более передают электрические сигналы одновременно, они накладываются и искажаются. Концентратор просто повторяет все полученные электрические сигналы, даже если он получает несколько сигналов одновременно. Рис. 2.18 демонстрирует ситуацию, когда компьютеры Арчи и Боба передают электрический сигнал одновременно (на этапах 1A и 1B), а концентратор повторяет оба сигнала для компьютера Ларри (этап 2).

Рис. 2.18. Коллизия, вызванная поведением концентратора LAN

ВНИМАНИЕ!

Концентратор передает каждый фрейм на все другие порты (кроме входящего). Так, фрейм компьютера Арчи поступит и на компьютер Ларри, и на компьютер Боба; а фрейм Боба — на компьютеры Ларри и Арчи.

Если концентратор на рис. 2.18 заменить коммутатором LAN, коллизия слева будет предотвращена. Коммутатор работает как устройство уровня 2, а значит, проверяет заголовок и концевик канала связи. Коммутатор просмотрел бы MAC-адреса, и даже если бы оба фрейма следовало передать на компьютер Ларри, то, передавая один фрейм, коммутатор поставил бы другой фрейм в очередь до завершения передачи первого.

Теперь вернемся к проблеме, созданной логикой концентратора, — *коллизии* (collision). Чтобы предотвратить коллизию, узлы Ethernet должны использовать полудуплексную логику вместо дуплексной. Проблема возникает только тогда, когда два устройства или более передают одновременно; полудуплексная логика требует от узлов подождать с передачей, если передает кто-то еще.

Предположим, например (см. рис. 2.18), что компьютер Арчи начал передавать фрейм чуть раньше, чем Боб попытался передать собственный. Получив первые биты этого фрейма, компьютер Боба на этапе 1B заметил бы, что кто-то еще начал передачу, и, согласно полудуплексной логике, он просто подождал бы с передачей своего фрейма.

Узлы, использующие полудуплексную логику, фактически используют относительно известный алгоритм CSMA/CD (Carrier Sense Multiple Access With Collision Detection — *множественный доступ с контролем несущей и обнаружением конфликтов*). Алгоритм заботится не только об очевидных случаях, но также и о проблемах, вызванных неудачной синхронизацией. Например, два узла могли бы одновременно проверить поступление фрейма, оба понимают, что никакой другой узел ничего не передает, и оба одновременно посылают свои фреймы, вызывая коллизию. Алгоритм CSMA/CD учитывает и разрешает такие случаи следующим образом.

Этап 1 Устройство, пересылающее фрейм, ожидает, пока не освободится среда Ethernet

Этап 2 Когда среда Ethernet свободна, отправитель (или отправители) начинает отправку фрейма

Этап 3 Отправитель прослушивает среду на предмет возникновения коллизии. Коллизия может произойти по многим причинам, включая неудачную синхронизацию. Если коллизия происходит, все передающие в настоящий момент узлы предпринимают следующее.

A. Посылают *сигнал оповещения о коллизии* (jamming signal), уведомляющий все узлы о ее возникновении.

B. Независимо выбирают случайный интервал времени ожидания перед повторной попыткой, чтобы избежать неудачной синхронизации.

C. Следующая попытка снова начинается с этапа 1.

Хотя большинство современных локальных сетей не часто использует концентраторы, а потому не нуждается в полудуплексном режиме, существует достаточно много старых корпоративных сетей, в которых концентраторы еще используются, поэтому необходимо быть готовым к решению задач на дуплекс. Каждый порт сетевой платы и коммутатора обладает дуплексным режимом. Он используется для всех каналов связи между компьютерами и коммутаторами или между коммутаторами. Но для любого канала связи, подключенного к концентратору LAN, соответствующий порт коммутатора или сетевой платы должен использовать полудуплексный режим. Обратите внимание, что сам концентратор не использует полудуплексную логику, он только повторяет входящий сигнал на всех портах.

На рис. 2.19 приведен пример с дуплексными каналами связи (*слева*) и одним концентратором LAN (*справа*). Концентратор требует, чтобы интерфейс F0/2 коммутатора SW2, наряду с подключенными к концентратору компьютерами, использовал полудуплексную логику.

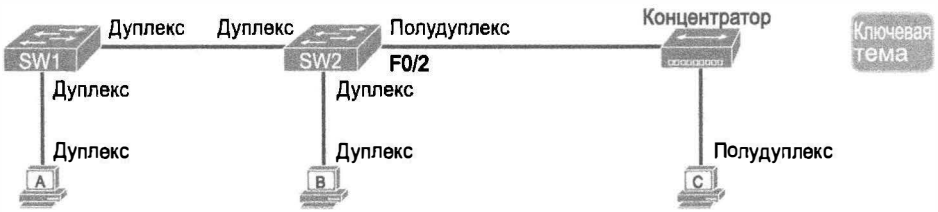


Рис. 2.19. Примеры использования интерфейсов с полным и полудуплексным режимами

Обзор

Резюме

- Обычно локальные сети объединяют соседние устройства, находящиеся в той же комнате, том же здании или в районе.
- За прошедшие годы разработано много типов локальных сетей, сейчас популярны два типа: локальные сети Ethernet и беспроводные локальные сети.
- Поскольку локальные сети Ethernet используют для каналов связи между узлами кабели, а в большинстве типов кабелей используются медные провода, такие сети зачастую называют *проводными локальными сетями*. Беспроводные локальные сети, напротив, не используют ни провода, ни кабели; вместо этого для каналов связи между узлами они используют радиоволны.
- Термин *Ethernet* относится к семейству стандартов LAN, которое совместно определяет физический и канальный уровни наиболее популярной проводной технологии LAN в мире. Стандарты, определенные Институтом инженеров по электротехнике и электронике (IEEE), определяют кабельную проводку, разъемы на концах кабелей, правила протоколов и все остальное, необходимое для создания локальной сети Ethernet.
- Стандарт Ethernet определяет только кабельную технологию LAN; другими словами, локальные сети Ethernet используют кабели.
- Все стандарты Ethernet изданы IEEE и содержат число 802.3 в начальной части названия стандарта.
- Фундаментальным критерием для выбора кабеля является материал его проводов, используемых для физической передачи битов: это либо медь, либо оптическое волокно.
  - Неэкранированная витая пара (UTP) дешевле оптоволоконного кабеля. Для передачи данных между узлами Ethernet такой кабель использует электрические провода.
  - Оптоволоконный кабель дороже, он передает данные в виде светового потока, распространяющегося внутри стеклянного оптического волокна в сердцевине кабеля.
- Технология Ethernet действует как единое целое, поскольку она использует тот же стандарт канального уровня для всех типов физических каналов связи.
- Термин *канал связи Ethernet* относится к любому физическому кабелю между двумя узлами Ethernet.
- Кабель UTP содержит набор медных проводов, сгруппированных в витые пары. Стандарты 10BASE-T и 100BASE-T требуют двух пар проводов, а стандарт 1000BASE-T — четырех.
- Большинство кабелей UTP имеют на обоих концах разъемы RJ-45. Разъем RJ-45 имеет восемь физических контактов, или контактных площадок, к которым могут быть прикреплены восемь проводов кабеля. Эти контакты обес-

печивают передачу электрического тока между концами медных проводов кабеля и электроникой узла.

- Протокол канала связи Ethernet определяет формат фрейма Ethernet: сначала заголовок Ethernet в середине передаваемых данных и в конце концевик Ethernet.
- Адреса Ethernet, называемые также адресами контроля доступа к среде передачи, или MAC-адресами, являются двоичными числами длиной 6 байтов (48 бит).
- Для удобства на большинстве компьютеров MAC-адреса представлены как шестнадцатеричные числа с 12-ю цифрами.
- При выпуске всем устройствам Ethernet присваивают всемирно уникальный MAC-адрес.
  - Первая часть MAC-адреса — это всемирно уникальный 3-байтовый код — *уникальный идентификатор организации* (OUI), присвоенный изготовителю IEEE.
  - Последние 3 байта числа изготовитель присваивает сам и никогда не использует их с тем же OUI.
  - В результате MAC-адрес каждого устройства уникален.
  - Адреса Ethernet известны под многими названиями: адрес LAN, адрес Ethernet, аппаратный адрес, встроенный адрес, физический адрес, универсальный адрес или MAC-адрес.
  - Поле FCS (Frame Check Sequence — контрольная сумма фрейма) — единственное поле в концевике Ethernet — позволяет получающему узлу сравнить полученный результат с отправленным и выяснить, не произошло ли ошибки.
  - Современные коммутаторы позволяют использовать дуплексную логику передачи, которая намного быстрее и проще полудуплексной логики, присущей концентраторам.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. В локальной сети малого офиса некоторые пользовательские устройства подключены к сети LAN по кабелю, а другие по беспроводной технологи (без кабеля). Что истинно из следующего, если в локальной сети используется Ethernet?

   А) Ethernet используют только устройства, подключенные по кабелю.

   Б) Ethernet используют только устройства, подключенные по беспроводной технологии.

   В) Ethernet используют все устройства, и подключенные по кабелю и по беспроводной технологии.

   Г) Ни одно из устройств не использует Ethernet.

2. Какой из следующих стандартов Ethernet определяет передачу трафика Gigabit Ethernet по кабелю UTP?

   А) 10GBASE-T.

   Б) 100BASE-T.

   В) 1000BASE-T.

   Г) Ни один из указанных выше ответов не верный.

3. Какое из перечисленных ниже утверждений о перекрещенном кабеле Fast Ethernet верно?

   А) Контакты 1 и 2 меняются местами на втором конце кабеля.

   Б) Контакты 1 и 2 на одном конце кабеля соединяются с контактами 3 и 6 на другом конце кабеля.

   В) Контакты 1 и 2 на одном конце кабеля соединяются с контактами 3 и 4 на другом конце кабеля.

   Г) Длина кабеля может достигать 1000 метров в каналах между зданиями.

   Д) Ни один из указанных выше ответов не верный.

4. Каждый вариант ответа описывает два различных устройства в сети, соединяемых кабелем 100BASE-T. Если эти устройства подключаются с помощью кабеля UTP, какие пары устройств требуют использования прямого кабеля? (Выберите три ответа.)

   А) Персональный компьютер и маршрутизатор.

   Б) Персональный компьютер и коммутатор.

   В) Коммутатор и концентратор.

   Г) Маршрутизатор и концентратор.

   Д) Беспроводная точка доступа (порт Ethernet) и коммутатор.

5. Какое из перечисленных ниже утверждений верно об алгоритме CSMA/CD?

   А) Алгоритм предупреждает коллизии.

   Б) Коллизия может произойти, но алгоритм определяет процесс уведомления компьютеров о возникновении коллизии и восстановления после нее.

   В) Алгоритм рассчитан только на два устройства в одном сегменте Ethernet.

   Г) Все перечисленные выше ответы ошибочны.

6. Что из перечисленного ниже верно о поле контрольной суммы во фрейме Ethernet?

   А) Это поле используется для восстановления информации при ошибках.

   Б) Длина этого поля равна 2 байтам.

   В) Это поле относится к концевику фрейма, а не заголовку.

   Г) Это поле используется для шифрования данных.

7. Что из перечисленного ниже верно о формате адреса Ethernet? (Выберите три ответа.)

   А) Каждый производитель помещает уникальный код OUI в первые 2 байта адреса.

   Б) Каждый производитель помещает уникальный код OUI в первых 3 байта адреса.

В) Каждый производитель помещает уникальный код OUI в первой половине адреса.

Г) Часть адреса, содержащая код производителя платы, называется MAC.

Д) Часть адреса, содержащая код производителя платы, называется OUI.

Е) Часть адреса, содержащая код производителя платы, не имеет определенного названия.

8. Какой из приведенных ниже терминов описывает адрес Ethernet, используемый для доставки фрейма более чем одному устройству в сети? (Выберите два ответа.)

А) Прошитый адрес (burned-in).

Б) Одноадресатный (unicast).

В) Широковещательный (broadcast).

Г) Многоадресатный (multicast).

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы приведены в табл. 2.5.

Таблица 2.5. Ключевые темы главы 2

Рис. 2.3	Типичные проводная и беспроводная корпоративные локальные сети	90
Табл. 2.1	Некоторые из типов локальных сетей Ethernet	91
Рис. 2.9	Принцип передачи по более чем двум электрическим каналам (каждый в одном направлении) между двумя узлами Ethernet	96
Рис. 2.10	Схемы расположения выводов прямого кабеля Ethernet на 10 и 100 Мбит/с	97
Рис. 2.12	Схемы расположения выводов перекрещенного кабеля Ethernet на 10 и 100 Мбит/с	98
Табл. 2.2	Устройства, передающие на проводной паре 1,2 и паре 3,6	98
Рис. 2.13	Пример интерфейсов, использующих полный и полудуплексный режимы	99
Рис. 2.15	Формат MAC-адресов Ethernet	102
Список	Определения полудуплексного и дуплексного режимов	105
Рис. 2.19	Примеры использования интерфейсов с полным и полудуплексным режимами	107

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

Ethernet, IEEE, проводная LAN (wired LAN), беспроводная LAN (wireless LAN), фрейм Ethernet (Ethernet frame), 10BASE-T, 100BASE-T, 1000BASE-T, Fast Ethernet, Gigabit Ethernet, канал Ethernet (Ethernet link), RJ-45, порт Ethernet (Ethernet port), плата сетевого интерфейса (Network Interface Card — NIC), прямой кабель (straight-through cable), перекрещенный кабель (crossover cable), адрес Ethernet (Ethernet address), MAC-адрес (MAC address), одноадресатный адрес (unicast address), широковещательный адрес (broadcast address), контрольная сумма фрейма (Frame Check Sequence — FCS)

Ответы на контрольные вопросы:
1 А. **2** В. **3** Б. **4** Б, Г и Д. **5** Б. **6** В. **7** Б, В и Д. **8** В и Г.

Глава 3

Основы сетей WAN

Большинство сетевых технологий уровня 1 и 2 относится к одной из двух основных категорий: распределенные сети (WAN) и локальные сети (LAN). Поскольку и глобальные, и локальные сети соответствуют уровням 1 и 2 модели OSI, у них много сходств: обе определяют подробности кабельных соединений, скоростей передачи, кодировки, способов передачи данных по физическим каналам связи, а также логику передачи фреймов по каналу связи.

Конечно, у глобальных и локальных сетей есть много различий: в частности, расстояние между узлами и бизнес-модель оплаты за услуги. По первому различию небольшую подсказку дают термины дистанции, *локальная* и *глобальная*: локальные сети объединяют соседние устройства, а глобальные — устройства, которые могут располагаться довольно далеко друг от друга, потенциально за сотни и даже тысячи километров.

Второе различие между этими сетями — оплата. За локальную сеть приходится платить самому, а глобальные сети сдаются в аренду. Для локальной сети нужно купить кабели и коммутаторы, а затем установить их в своих помещениях. Глобальные сети физически проложены по собственности других людей, где вы не имеете права протянуть свои кабели и расположить устройства. Этим, как правило, занимается несколько компаний, телефонных или кабельных, они имеют собственные устройства и кабели, создают собственные сети, а затем предоставляют право за плату передавать данные по своим сетям.

Три основных раздела этой главы знакомят с глобальными сетями. В первом речь пойдет о выделенной линии глобальной сети — типе канала связи WAN, используемого корпоративными сетями с 1960-х годов. Во втором будет продемонстрировано применение технологии Ethernet для создания служб WAN, а также описаны преимущества более длинных кабелей современных оптоволоконных кабелей. И в третьем разделе будут представлены популярные технологии WAN, используемые для доступа к Интернету.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы.

Выбор компонентов сети, удовлетворяющих заданной спецификации.

Передача данных между двумя хостами по сети.

Основные темы

Выделенные линии сетей WAN

Предположим, вы главный инженер объединенной корпоративной сети TCP/IP. Ваша компания строит новое здание на расстоянии 100 км от своей штаб-квартиры. Вы, конечно, проложите локальную сеть по новому зданию, но эту новую локальную сеть придется соединить с остальной частью существующей корпоративной сети TCP/IP.

Для подключения локальной сети нового здания к остальной части существующей корпоративной сети необходима сеть WAN, которая как минимум должна передавать данные между дистанционной локальной сетью и остальной частью существующей сети. Именно это и делают выделенные линии сети WAN — передают данные между двумя маршрутизаторами.

В общем виде выделенная линия WAN действует как перекрещенный кабель Ethernet, соединяющий два маршрутизатора, но почти безграничной длины. Выделенная линия обеспечивает дуплексный обмен данными между любыми маршрутизаторами на расстоянии десятков, сотен и даже тысяч километров.

Этот раздел начинается с обсуждения положения выделенных линий относительно локальных сетей и маршрутизаторов, поскольку одной из главных задач сетей WAN является передача данных между локальными сетями. Остальная часть раздела посвящена физическим подробностям выделенных линий и информации о протоколах канала связи.

Расположение выделенных линий относительно локальных сетей и маршрутизаторов

Подавляющее большинство устройств конечного пользователя в корпоративной или домашней сети подключено непосредственно к локальной сети. Для соединения с коммутатором большинство компьютеров использует сетевую плату Ethernet. Ныне все больше устройств, таких как телефоны и планшеты, используют беспроводные локальные сети 802.11, поддерживающие только беспроводные соединения LAN.

Теперь представьте типичную компанию со множеством разных мест. С точки зрения человеческих ресурсов у компании может быть много сотрудников, работающих в множестве разных мест. С точки зрения помещений у компании может быть несколько больших площадок, с сотнями или тысячами отдельных филиалов, хранилищ и других мест. Но с точки зрения сети каждую площадку можно считать одной или несколькими локальными сетями, которые должны общаться друг с другом, а для этого они должны быть соединены друг с другом при помощи WAN.

Для соединения локальных сетей применяют сеть WAN — объединенную сеть, использующую маршрутизаторы, подключенные к каждой из локальных сетей, и канал связи между ними. В первую очередь сетевой инженер предприятия заказал бы некий канал связи WAN. Маршрутизатор на каждой площадке подключается и к каналу связи WAN, и к локальной сети (LAN), как показано на рис. 3.1. Обратите внимание на зигзагообразную линию между маршрутизаторами — это общепринятый способ обозначения выделенной линии, когда на рисунке не нужно изображать ее физические детали.

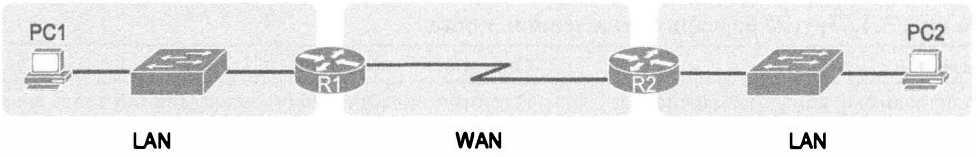


Рис. 3.1. Сеть малого предприятия с одной выделенной линией

Кроме представленной на рисунке выделенной линии, мир технологий WAN богат многими разными возможностями. Технология WAN объединяет многочисленные типы физических каналов связи, а также контролирующих их протоколов канала связи. В мире проводных локальных сетей, напротив, осталась в основном только одна главенствующая технология — Ethernet, поскольку она выиграла сражение на рынке проводных локальных сетей в 1980–1990-х годах.

Физические детали выделенных линий

Служба выделенной линии передает биты в обоих направлениях на заранее определенной скорости в дуплексном режиме. Концептуально он действует как дуплексный перекрестный кабель Ethernet между двумя маршрутизаторами (рис. 3.2). Выделенная линия использует две пары проводов, по одной для каждого направления передачи данных, чтобы обеспечить дуплексный режим.

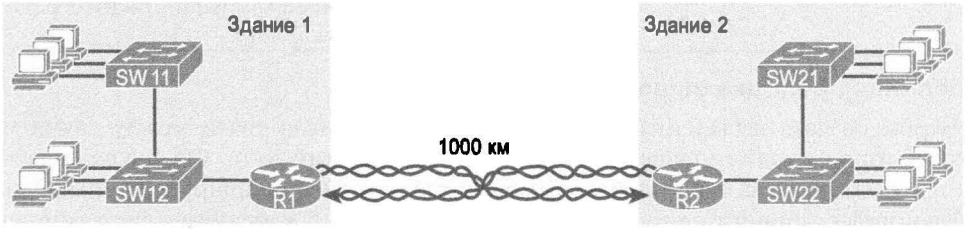


Рис. 3.2. Концептуальное представление службы выделенной линии

Конечно, у выделенных линий много отличий от перекрещенных кабелей Ethernet. Чтобы создать канал связи такой длины, выделенная линия не может фактически быть единым длинным кабелем между двумя площадками. Но телефонные компании имеют большие разветвленные сети телефонных кабелей и специализированных коммутационных устройств, позволяющих создать их собственные компьютерные сети. Сеть телефонной компании действует как перекрещенный кабель между двумя точками, но физическая действительность скрывается от клиента.

Выделенные линии имеют собственную терминологию. Термин *выделенная, или арендованная, линия* свидетельствует о том факте, что использующая выделенную линию компания не владеет каналом, а вносит ежемесячную арендную плату за его использование. Сейчас популярен термин *провайдер услуг* (service provider), обозначающий компанию, предоставляющую все формы подключения WAN, включая службы Интернета.

С учетом столь давней истории у выделенных линий было много названий. Список некоторых из них вместе с кратким описанием приведен в табл. 3.1.

Таблица 3.1. Другие названия выделенной линии

Название	Описание
Арендованный канал (leased circuit), канал (circuit)	В терминологии телефонных компаний слова *линия* (line) и *канал* (circuit) используются как синонимы и означают электрический канал между двумя конечными точками
Последовательный канал связи (serial link), последовательная линия (serial line)	Слова *канал связи* (link) и *линия* (line) также используются как синонимы. *Последовательный* (serial) в данном случае свидетельствует о том факте, что биты передаются последовательно и маршрутизаторы используют последовательные интерфейсы
Двухточечный канал связи (point-to-point link), двухточечная линия (point-to-point line)	Означает, что топология растянута между двумя и только двумя точками. (Некоторые устаревшие выделенные линии допускали более двух устройств.)
Канал T1	Специальный тип выделенной линии, передающей данные на скорости 1,544 Мбит/с
Канал связи WAN (WAN link), канал связи (link)	Очень общие термины, не указывающие ни какой конкретной технологии
Частная линия (private line)	Свидетельствует о том факте, что посланные по линии данные не могут быть скопированы другими клиентами телефонной компании. Таким образом, данные являются частными

Кабельная проводка выделенной линии

Чтобы создать выделенную линию, на концах канала связи между двумя маршрутизаторами должен существовать некий физический путь. Физическая кабельная проводка должна соединять оба здания, где находятся маршрутизаторы. Однако телефонной компании не нужно прокладывать единый кабель между этими двумя зданиями — она использует большую и сложную сеть, которая создает видимость кабеля между этими двумя маршрутизаторами.

Некоторое представление о кабельной проводке, которая могла бы существовать в телефонной компании для создания короткой выделенной линии, дает рис. 3.3. Телефонная компания размещает свое оборудование в зданиях *центральных станций* (Central Office — CO) и прокладывает кабели от станций до всех зданий в городе, предполагая предоставлять свои услуги людям в этих зданиях. Затем телефонная компания настраивает свои коммутаторы так, чтобы использовать часть пропускной способности на каждом кабеле для передачи данных в обоих направлениях, создавая эквивалент перекрещенного кабеля между двумя маршрутизаторами.

Хотя происходящее в телефонной компании полностью скрыто от ее клиента, инженеры компании должны знать о находящихся в здании клиента элементах канала связи, расположенных перед маршрутизатором.

На каждой площадке установлено *клиентское оборудование* (Customer Premises Equipment — CPE), включающее маршрутизатор, плату последовательного интерфейса и модуль CSU/DSU. Каждый маршрутизатор использует плату последовательного интерфейса, действующую подобно сетевой плате Ethernet, посылающей и получающей данные по физическому каналу связи. Физическому каналу связи требуется *модуль обслуживания канала и данных* (Channel Service Unit/Data Service

Unit — CSU/DSU). Модуль CSU/DSU может быть интегрирован в плату последовательного интерфейса маршрутизатора или представлять собой внешнее устройство. На рис. 3.4 представлены устройства CPE, а также кабельная проводка.

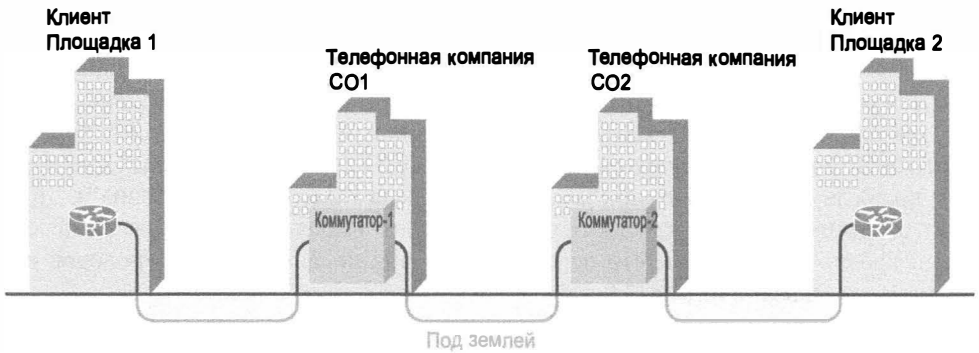


Рис. 3.3. Возможная кабельная проводка телефонной компании для короткой выделенной линии

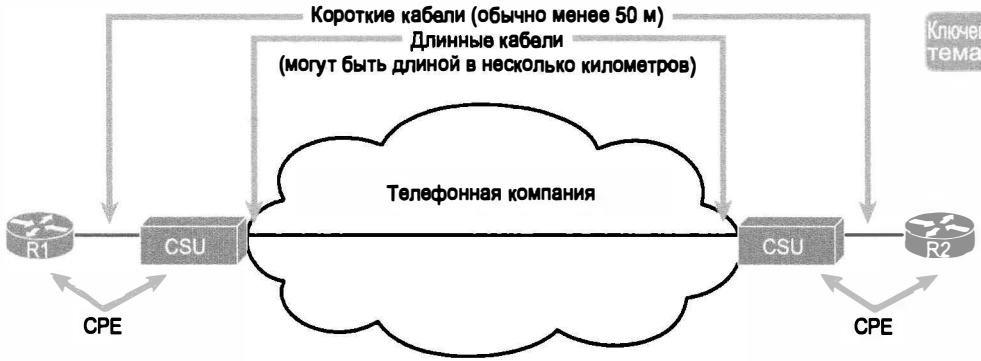


Рис. 3.4. Типичная схема кабельной проводки CPE для выделенной линии

Кабельная проводка включает короткий последовательный кабель (только если используется внешний модуль CSU/DSU) и кабель телефонной компании, выделенный для самой линии. Последовательный кабель соединяет последовательный интерфейс маршрутизатора с внешним модулем CSU/DSU. (Существуют много видов кабелей; кабель должен только подходить к разъему последовательного интерфейса на одном конце и к модулю CSU/DSU на другом.) Как правило, четырехжильный кабель от телефонной компании подключается к модулю CSU/DSU через разъем RJ-48, имеющий тот же размер и форму, что и разъем RJ-45 (см. рис. 2.7 в главе 2).

Телефонные компании предоставляют для выделенных линий широкое разнообразие скоростей передачи. Клиент не может заказать точную скорость по своему желанию, он должен выбрать из длинного списка предопределенных скоростей. Медленные каналы связи работают на скоростях, кратных 64 Кбит/с, а более быстрые — на скоростях, кратных примерно 1,5 Мбит/с.

Создание канала связи WAN в лабораторных условиях

При подготовке к экзаменам CCENT и CCNA можно купить бывшие в употреблении маршрутизатор и коммутатор для профессиональной практики. Так можно

создать эквивалент выделенной линии, даже не имея реальной выделенной линии от телефонной компании и блоков CSU/DSU. В этом коротком разделе предоставлено достаточно информации для создания канала связи WAN в лабораторных или домашних условиях.

Последовательные кабели, соединяющие маршрутизатор с внешним модулем CSU/DSU, называются кабелями *терминального оборудования* (Data Terminal Equipment — DTE). Для создания физического канала связи WAN в лабораторных условиях необходимы два последовательных кабеля: последовательный кабель DTE и подобный, но немного другой кабель *аппаратуры передачи данных* (Data Communications Equipment — DCE). У кабеля DCE есть гнездо ("мама"), а у кабеля DTE — разъем ("папа"), что позволяет соединить эти два кабеля непосредственно. Кабель DCE похож на перекрещенный кабель Ethernet тем, что передающая и принимающая пары проводов перекрещены, как показано на рис. 3.5.

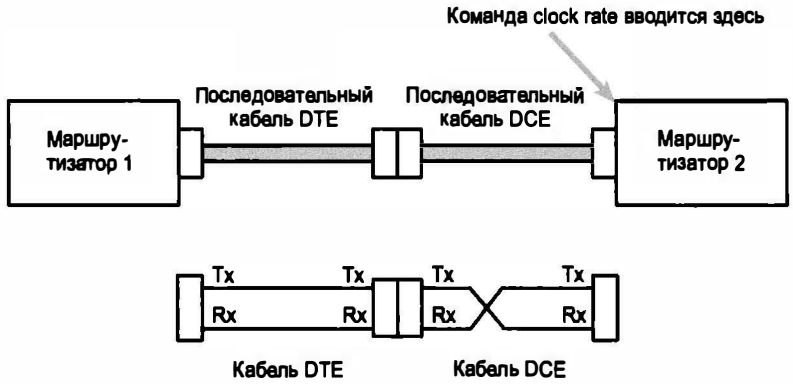


Рис. 3.5. Последовательные кабельные соединения, использующие кабели DTE и DCE

На рис. 3.5, *сверху*, представлено соединение кабелей, а *снизу* — их внутренняя конструкция. Обратите внимание на последовательный кабель DTE, который действует как прямой кабель, не меняя местами передающую и принимающую пары, в то время как кабель DCE перекрещивает их.

И наконец, чтобы канал связи заработал, маршрутизатор с установленным кабелем DCE должен выполнять функцию, обычно выполняемую блоком CSU/DSU. Блок CSU/DSU обычно обеспечивает *синхронизацию* (clocking), указывая маршрутизатору, когда именно передавать по последовательному кабелю каждый бит. Последовательный интерфейс маршрутизатора способен обеспечивать синхронизацию, но маршрутизатор не делает этого без команды **`clock rate`**. Пример необходимой настройки конфигурации приведен в главе 15.

Подробности о канале связи выделенных линий

Выделенная линия представляет службы уровня 1. Другими словами, она обещает доставлять биты между устройствами, соединенными выделенной линией. Однако сама выделенная линия не определяет протокол канального уровня, который будет использоваться на ней.

Поскольку выделенные линии определяют только службу передачи уровня 1, большинство компаний и организаций по стандартизации выработали для их управления протоколы канала связи. Ныне наиболее популярными протоколами канального уровня, используемыми для выделенных линий между двумя маршрутизаторами, являются *высокоуровневый протокол управления каналом* (High-Level Data Link Control — HDLC) и *протокол двухточечного соединения* (Point-to-Point Protocol — PPP). Протокол HDLC кратко рассматривается в следующем разделе, только чтобы продемонстрировать один пример, а также дано несколько замечаний о том, как маршрутизаторы используют протоколы канала связи WAN.

Основы протокола HDLC

Все протоколы канала связи выполняют подобную задачу: контролируют правильную передачу данных по физическому каналу связи определенного типа. Например, для идентификации устройства, которое должно получить данные, протокол канала связи Ethernet использует поле адреса получателя, а поле FCS позволяет принимающему устройству проверить правильность полученных данных. Протокол HDLC обеспечивает подобные функции.

Благодаря простой двухточечной топологии сети у протокола HDLC выделенной линии меньше работы. Когда один маршрутизатор посылает фрейм HDLC, он может поступить только в одно место: на другой конец канала связи. Поэтому, несмотря на наличие поля адреса, получатель предопределен. Когда я обедаю с другом Гари наедине, я не должен начинать каждую фразу со слов "Эй, Гари", он и так знает, что я обращаюсь к нему.

ВНИМАНИЕ!

Если интересно, почему у протокола HDLC вообще есть поле адреса, то в прошлом телефонные компании предоставляли также и многоточечные каналы. Эти каналы соединяли более двух устройств, поэтому был возможен более чем один получатель, что требовало для его идентификации наличия поля адреса.

Протокол HDLC обладает и другими полями и функциями, также подобными Ethernet. В табл. 3.2 приведен список полей протокола HDLC, а также подобных полей заголовка и концевика протокола Ethernet.

Таблица 3.2. Сравнение полей заголовка HDLC и Ethernet

Поле заголовка или концевика HDLC	Эквивалент Ethernet	Описание
Флаг (flag)	Преамбула (preamble)	Содержит распознаваемую битовую схему, которая дает получающему узлу понять, что прибывает новый фрейм
Адрес (address)	Адрес получателя (destination address)	Идентифицирует устройство получателя
Тип (type)	Тип (type)	Идентифицирует тип пакета уровня 3, инкапсулируемого во фрейме
Контрольная сумма фрейма (FCS)	Контрольная сумма фрейма (FCS)	Используется для проверки фрейма на целостность и отсутствие ошибок; это единственное поле концевика в данной таблице

Ныне протокол HDLC утвержден как стандарт *Международной организации по стандартизации* (International Organization for Standardization — ISO), той же организацией, которая выработала модель OSI. Однако по стандарту ISO у протокола HDLC нет поля Type, и маршрутизаторы должны сами знать тип пакета во фрейме. Поэтому маршрутизаторы Cisco используют собственный вариант протокола HDLC, в который добавлено поле Type (рис. 3.6).

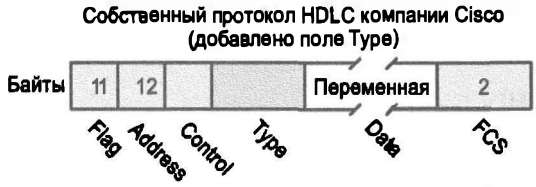


Рис. 3.6. Структура фрейма протокола HDLC

Как маршрутизаторы используют канал связи WAN

В настоящее время большинство выделенных линий подключено к маршрутизаторам, а маршрутизаторы сосредоточены на доставке пакетов хосту получателя. Однако физически маршрутизаторы подключены и к локальным, и к глобальным сетям, требующим передачи внутренних фреймов канала связи. Теперь, когда вы имеете некие знания о протоколе HDLC, рассмотрим, как маршрутизаторы используют его при отправке данных.

Сетевой уровень модели TCP/IP отвечает за перенаправление пакетов IP с хоста отправителя на хост получателя. Базовые локальные и глобальные сети действуют как средство передачи пакетов на следующий маршрутизатор или устройство конечного пользователя (рис. 3.7).

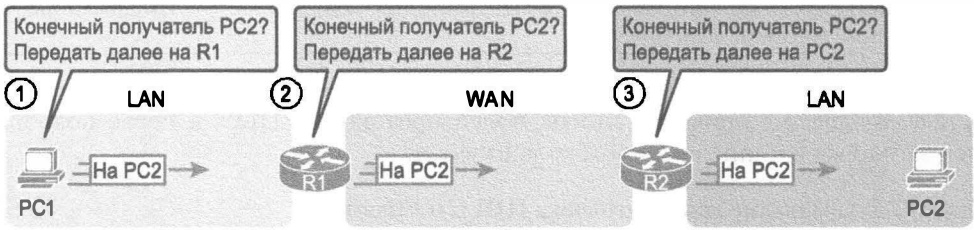


Рис. 3.7. Логика маршрутизации IP по локальным и глобальным сетям

Рассмотрим этапы следования пакета, переданного с компьютера PC1 на IP-адрес компьютера PC2.

1. Логика сетевого уровня (IP) компьютера PC1 требует посылать пакет на соседний маршрутизатор (R1).
2. Логика сетевого уровня маршрутизатора R1 требует перенаправить пакет на выделенную линию к маршрутизатору R2.
3. Логика сетевого уровня маршрутизатора R2 требует перенаправить пакет по каналу связи LAN на компьютер PC2.

Как показано на рис. 3.7, чтобы фактически переместить биты пакета, компьютеры и маршрутизаторы должны полагаться на логику сетевого уровня, локальные и глобальные сети. На рис. 3.8 приведена та же сеть с тем же пакетом, но на сей раз показана часть логики канального уровня, используемой хостами и маршрутизаторами. Три разных этапа канального уровня инкапсулируют пакет во фрейме канала связи на трех транзитных участках объединенной сети: от PC1 до R1, от R1 до R2 и от R2 до PC2.

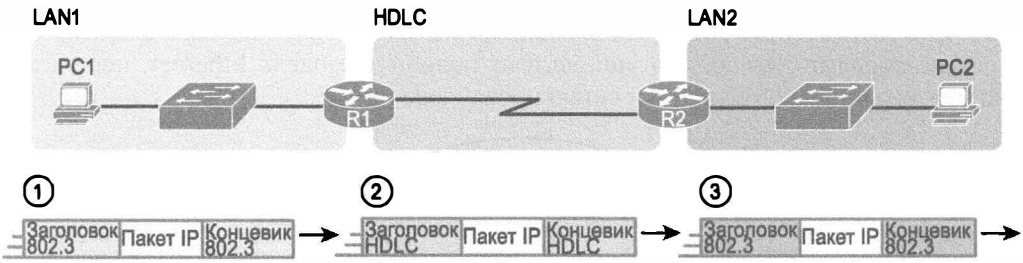


Рис. 3.8. Общая концепция деинкапсуляции и реинкапсуляции пакетов IP маршрутизаторами

Снова рассмотрим этапы следования пакета, переданного с компьютера PC1 на IP-адрес компьютера PC2.

1. Чтобы послать пакет IP на маршрутизатор R1, компьютер PC1 инкапсулирует его во фрейм Ethernet, обладающий MAC-адресом получателя (R1).
2. Маршрутизатор R1 деинкапсулирует (извлекает) пакет IP из фрейма Ethernet и инкапсулирует (помещает) его во фрейм HDLC, используя заголовок и концевик HDLC, а затем передает фрейм HDLC маршрутизатору R2.
3. Маршрутизатор R2 извлекает пакет IP из фрейма HDLC и помещает его во фрейм Ethernet, обладающий MAC-адресом получателя (PC2),а затем перенаправляет фрейм Ethernet на компьютер PC2.

В результате выделенная линия с протоколом HDLC создает канал связи WAN между двумя маршрутизаторами, чтобы они могли переправлять пакеты для устройств в локальных сетях на его концах. Сама выделенная линия представляет собой физические средства передачи битов в обоих направлениях. Фреймы HDLC — это средство инкапсуляции пакетов сетевого уровня при передаче по каналу связи между маршрутизаторами.

У выделенных линий много преимуществ, обеспечивших их относительно продолжительное присутствие на рынке WAN. Эти линии просты для клиента, широко доступны, имеют высокое качество и являются частными. Но у них есть и недостатки по сравнению с более новыми технологиями WAN, включая достаточно высокую стоимость и более продолжительное время выполнения заказа на установку службы.

В следующем разделе описана альтернативная технология WAN, используемая в некоторых примерах данной книги: Ethernet.

Ethernet как технология WAN

На протяжении нескольких первых десятилетий существования технология Ethernet применялась только для локальных сетей. Ограниченная длина кабелей (1-

2 км) вполне позволяла протянуть локальную сеть в пределах района, но это было пределом.

Со временем IEEE улучшил стандарты Ethernet так, что эта технология стала подходящей и для WAN. Например, стандарт 1000BASE-LX позволяет использовать *одномодовый оптоволоконный кабель* (single-mode fiber) длиной 5 километров, а стандарт 1000BASE-ZX даже 70. В результате технология Ethernet стала вполне приемлемой технологией WAN.

Сейчас, на втором десятилетии двадцать первого столетия, большинство *провайдеров услуг*(Service Provider — SP) предлагают службы WAN на базе Ethernet под многими именами. Но все они используют подобную модель Ethernet, используемую между площадкой клиента и сетью провайдера (рис. 3.9).

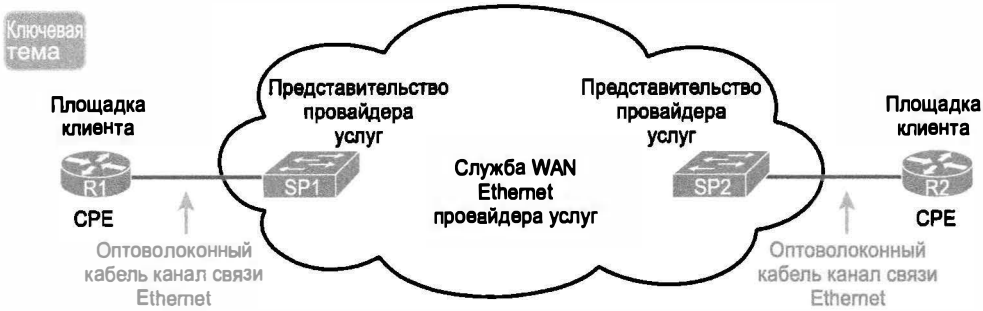


Рис. 3.9. Оптоволоконный канал связи Ethernet, соединяющий маршрутизатор CPE с глобальной сетью провайдера услуг

Большинство концепций на рис. 3.9 совпадает с выделенной линией телефонной компании, показанной на рис. 3.3, но теперь с каналами связи и устройствами Ethernet. Клиент подключается к каналу связи Ethernet через интерфейс маршрутизатора. Оптоволоконный кабель канала связи Ethernet соединяет здание клиента с ближайшим *представительством* (Point of Presence — PoP) провайдера услуг. Вместо коммутатора телефонной компании (см. рис. 3.3) SP использует коммутатор Ethernet. В сети провайдера могут использоваться любые технологии, необходимые для создания конкретной службы WAN Ethernet.

Глобальные сети Ethernet, поддерживающие службы уровня 2

Служба WAN, показанная на рис. 3.9, может включать широкое разнообразие служб с большим количеством сложных сетевых концепций, необходимых для ее построения. Поскольку это только третья глава вашей, вероятно, первой книги по сертификации Cisco, пока нет смысла углубляться в детали этих служб. Поэтому в целях сертификации CCENT эта книга сосредоточивается лишь на одной конкретной службе WAN — Ethernet, работу которой легко понять, если понятно, как работают локальные сети Ethernet.

Используемая для примеров сертификации CCENT и CCNA Routing and Switching служба WAN Ethernet известна под двумя названиями: *эмуляция Ethernet* (Ethernet emulation) и *Ethernet поверх MPLS* (Ethernet over MPLS — EoMPLS). Эмуляция Ethernet — это общий термин, означающий службу, действующую как один канал связи Ethernet. Служба EoMPLS относится к *мультипротокольной коммута-*

ции по меткам (Multiprotocol Label Switching — MPLS) — одной из технологий, применяемой в облаке (cloud) SP. В этой книге данная служба называется эмуляцией Ethernet, или EoMPLS.

ВНИМАНИЕ!

О широком разнообразии сетей провайдера услуг на рис. 3.9 и сертификации Cisco. У компании Cisco есть три сертификации (CCNA, CCNP и CCIE) во многих областях: маршрутизация и коммутация, голос, защита и т.д. Два направления (провайдер услуг и работа провайдера услуг) сосредоточены на технологиях и задачах провайдера услуг. Более подробная информация по этой теме приведена по адресу `www.cisco.com/go/certifications`.

Обсуждаемый в этой книге тип службы EoMPLS предоставляет клиенту канал связи Ethernet между двумя площадками, т.е. служба EoMPLS обеспечивает следующее:

- двухточечное соединение между двумя устройствами клиента;
- поведение, как будто между этими двумя устройствами существует оптоволоконный канал связи Ethernet.

Вообразите два маршрутизатора с одним каналом связи Ethernet между ними — это именно то, что делает служба EoMPLS.

Эта идея представлена на рис. 3.10. В данном случае это маршрутизаторы R1 и R2, подключенные к службе EoMPLS вместо последовательного канала связи. Маршрутизаторы используют интерфейсы Ethernet и могут передавать данные в обоих направлениях одновременно. Физически каждый маршрутизатор подключен к ближайшему представительству провайдера услуг, как было показано ранее на рис. 3.9, но логически эти два маршрутизатора просто могут послать фреймы Ethernet друг другу по каналу связи.

Рис. 3.10. Служба EoMPLS действует как простой канал связи Ethernet между двумя маршрутизаторами

Как маршрутизаторы перенаправляют пакеты IP, используя эмуляцию Ethernet

Глобальные сети по самой своей природе предоставляют маршрутизаторам IP способ маршрутизации пакетов IP от локальной сети на одной площадке по сети WAN к другой локальной сети на другой площадке. Маршрутизация по каналу связи EoMPLS все еще подразумевает использование сети WAN как способ перенаправления пакетов IP с одной площадки на другую. Однако канал связи WAN иногда использует те же протоколы Ethernet, что и каналы связи LAN на каждой площадке.

Канал связи EoMPLS использует Ethernet и для функций уровня 1, и уровня 2. Это означает, что канал связи использует те же, уже знакомые заголовок и концевик Ethernet, представленные на рис. 3.11, *посередине*.

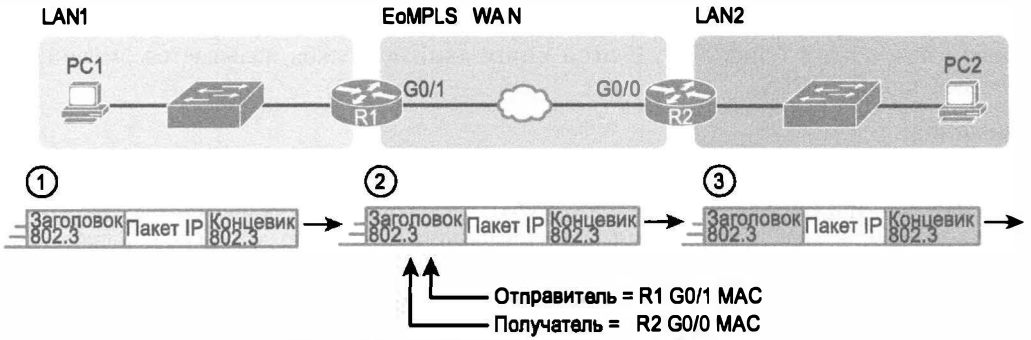


Рис. 3.11. Маршрутизация по каналу EoMPLS

ВНИМАНИЕ!

В этой книге соединения EoMPLS изображаются сплошной черной линией, как другие каналы связи Ethernet, но с небольшим облаком посередине, обращающем внимание на то, что это специфический канал связи Ethernet через службу WAN.

Здесь представлены те же три этапа маршрутизации, что и у последовательного канала связи на рис. 3.8. В данном случае все три этапа маршрутизации используют тот же протокол Ethernet (802.3). Но обратите внимание, что заголовок и концевик канала связи каждого фрейма разные. Каждый маршрутизатор отказывается от прежнего заголовка/концевика канала связи и добавляет новые, как показано на этих этапах. Обратите внимание на этап 2, поскольку этапы 1 и 3 остались неизменными по сравнению с показанными на рис. 3.8.

1. Чтобы послать пакет IP далее на маршрутизатор R1, компьютер PC1 инкапсулирует его во фрейм Ethernet, обладающий MAC-адресом получателя (R1).
2. Маршрутизатор R1 извлекает пакет IP из фрейма Ethernet и помещает его в новый фрейм Ethernet, с новым заголовком и концевиком Ethernet. MAC-адресом получателя будет MAC-адрес порта G0/0 маршрутизатора R2, а отправителя — MAC-адрес порта G0/1 маршрутизатора R1. Затем маршрутизатор R1 перенаправляют этот фрейм по службе EoMPLS маршрутизатору R2.
3. Маршрутизатор R2 извлекает пакет IP из фрейма HDLC и помещает его во фрейм Ethernet с MAC-адресом получателя PC2, а затем перенаправляет его на компьютер PC2.

Доступ к Интернету

Многие начинают подготовку к сертификации CCENT и CCNA, никогда не слышав о выделенных линиях, но о двух других технологиях WAN, используемых для доступа к Интернету, знают почти все, — это *цифровой абонентский канал* (Digital Subscriber Line — DSL) и *кабель* (cable). Эти две технологии WAN не заменяют выделенные линии во всех случаях, но играют важную роль в конкретном случае создания соединения WAN между домом или офисом и Интернетом.

Этот раздел начинается со знакомства с базовыми концепциями сетей Интернета, а также с некоторыми специфическими особенностями того, как канал DSL и кабель обеспечивают обмен данными с Интернетом.

Интернет как большая сеть WAN

Интернет — удивительное культурное явление. Большинство из нас используют его каждый день. Мы размещаем сообщения в социальных сетях, ищем информацию в поисковой системе Google и передаем электронную почту. Мы используем такие приложения мобильного телефона, как прогноз погоды, карты и обзоры по фильмам. Мы используем Интернет для покупки как физических товаров, так и загрузки цифровых продуктов — музыки и видео. Интернет создал совершенно новые вещи и изменил старый образ жизни прежних поколений людей.

Но если обратить внимание на сетевые технологии, лежащие в основе Интернета, то окажется, что это просто одна огромная сеть TCP/IP. Фактически свое название Интернет берет от базового протокола сетевого уровня: *протокола Интернета* (Internet Protocol). Интернет включает множество локальных сетей и охватывает весь земной шар, а потому, конечно же, нуждается в каналах связи WAN, соединяющих различные площадки.

Как сеть сетей Интернет фактически принадлежит бесчисленным компаниям и людям, включает в себя большинство корпоративных сетей TCP/IP, огромное количество домашних сетей, множество мобильных телефонов и других беспроводных устройств (рис. 3.12).

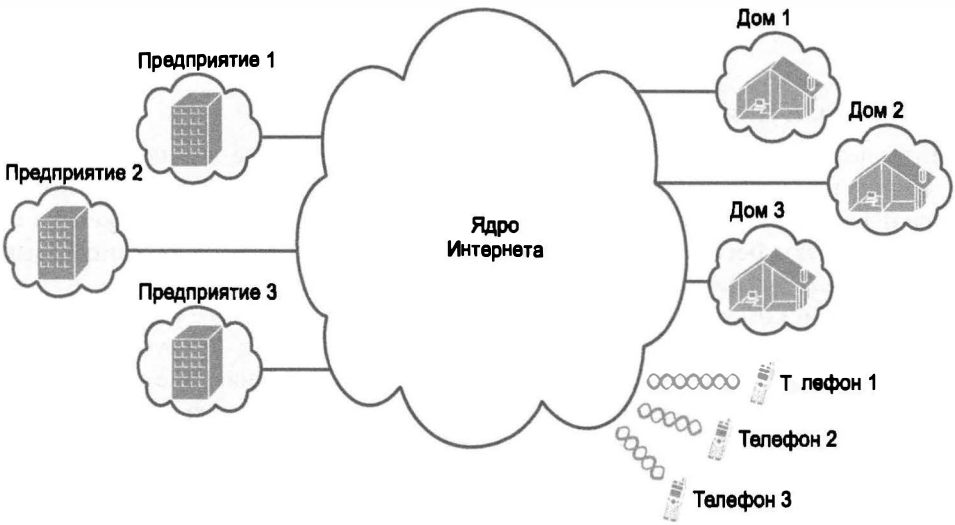


Рис. 3.12. Интернет с корпоративными, домашними и телефонными абонентами

Область на рис. 3.12, *посередине*, *ядро Интернета* (Internet core), представляет собой локальные и глобальные сети, принадлежащие *провайдерам услуг Интернета* (Internet Service Providers — ISP). (Как общепринято, ядро Интернета на рис. 3.12 изображено в виде облака, скрывающего детали этой части сети.) В ядре Интернета провайдеры ISP совместно формируют сеть каналов связи между собой так, чтобы,

независимо от того, через кого из них будут подключены конкретная компания или человек, всегда существовал путь на любое другое подключенное устройство.

На рис. 3.13 представлена схема, немного отличная от рис. 3.12, — в данном случае показана концепция ядра Интернета: сети ISP, соединенные со всеми своими клиентами, а также друг с другом так, чтобы пакеты IP могли передаваться каждым клиентом каждого ISP на любой клиент любого ISP.

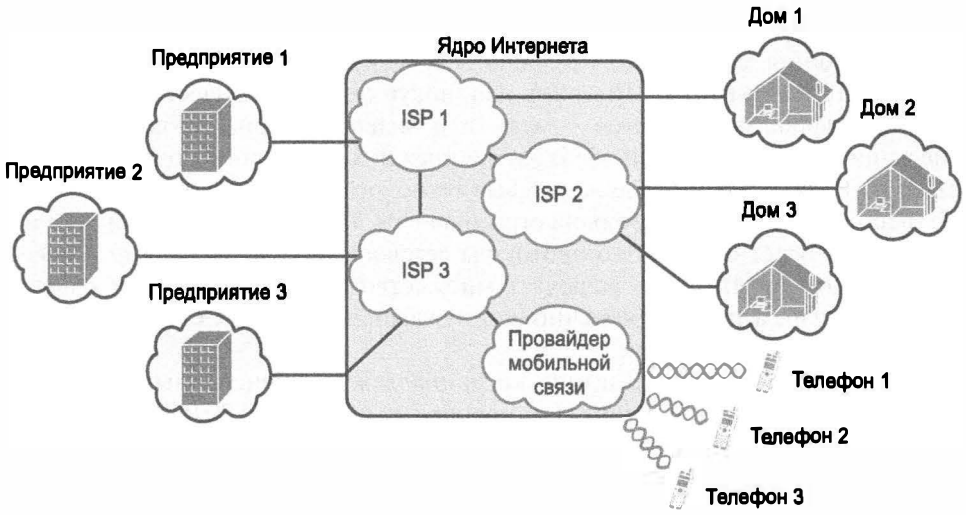


Рис. 3.13. Ядро Интернета с несколькими провайдерами услуг Интернета и телефонными компаниями

Каналы связи для доступа к Интернету

Интернет использует множество каналов связи WAN. Все линии, соединяющие предприятия и дома с одним из провайдеров услуг Интернета на рис. 3.13, представляют некий кабельный канал связи WAN, а телефоны для своего канала связи WAN используют беспроводную технологию. Эти каналы связи обычно называют *каналом связи Интернета* (Internet access link).

Исторически компании предпочитают использовать в качестве каналов связи Интернета набор технологий WAN, а домашние пользователи — другие технологии. Предприятия нередко используют выделенные линии, соединяющие их маршрутизатор с маршрутизатором провайдера ISP. Такой пример приведен на рис. 3.14, *сверху*.

Пользователи каналов связи Интернета зачастую используют такие технологии, как DSL и кабель. Эти технологии используют кабельную проводку, уже имеющуюся в большинстве домов, что существенно удешевляет подключение домашних пользователей. Технология DSL использует существующие аналоговые телефонные линии, а кабельный Интернет — телевизионные кабели.

ВНИМАНИЕ!

Хотя технологии DSL и кабель используют главным образом домашние потребители, предприятия также вполне могут их использовать.

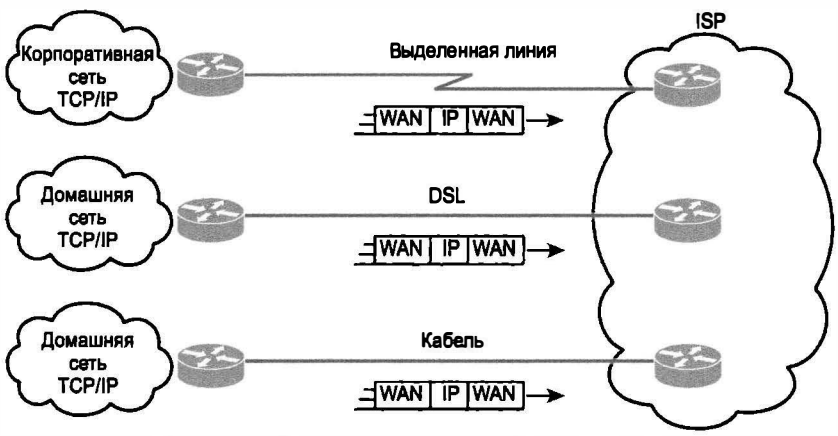


Рис. 3.14. Три примера каналов связи Интернета

Все три технологии доступа к Интернету, показанные на рис. 3.14, используют два маршрутизатора: один на стороне клиента и один на стороне ISP. Передавая по назначению пакеты IP и перенаправляя их следующему маршрутизатору, маршрутизаторы продолжают использовать логику сетевого уровня. Однако физические детали и детали канального уровня канала связи WAN отличаются от таковых у выделенных линий. Некоторые из этих различий рассматриваются ниже.

Цифровой абонентский канал

Цифровой абонентский канал (Digital Subscriber Line — DSL) позволяет создать относительно короткий (километры, но не десятки километров) высокоскоростной канал связи WAN между клиентом телефонной компании и ISP. Для него используется обычная домашняя телефонная линия из пары проводов. Технология DSL не предназначена для замены выделенных линий между любыми двумя площадками на потенциально очень большом расстоянии. Она обеспечивает довольно короткий физический канал связи от дома до сети телефонной компании, предоставляя доступ к Интернету.

Чтобы лучше разобраться в кабельной проводке, рассмотрим типичную домашнюю телефонную сеть, существовавшую до появления DSL. У каждого дома есть одна телефонная линия, проложенная до ближайшей телефонной станции. Как показано на рис. 3.15, *слева*, телефонные провода разделяются и выводятся на несколько стенных розеток, обычно с гнездами под разъем RJ-11, который немного уже похожего разъема RJ-45.

Теперь рассмотрим телефонную линию и оборудование телефонной станции. В свое время телефонная компания проложила телефонные линии от телефонных станций до всех окрестных домов. На телефонной станции каждая линия соединена с портом на коммутаторе. Этот коммутатор обеспечивает возможность вызова, соединения и передачи голоса по международной голосовой сети — *коммутируемой телефонной сети общего пользования* (Public Switched Telephone Network — PSTN).

Для установки службы DSL в доме на рис. 3.15 следует установить устройства DSL и дома, и на телефонной станции. Совместно оборудование DSL на каждой стороне локальной телефонной линии способно одновременно отправлять данные и поддерживать голосовой трафик.

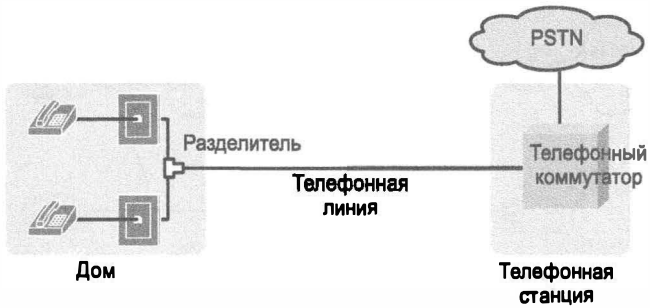


Рис. 3.15. Концепция телефонной сети в США

На рис. 3.16, *слева*, представлены изменения в доме: ко второй розетке подключен новый *модем DSL* (DSL modem). Для обмена данными с телефонной компанией модем DSL удовлетворяет физическим стандартам и стандартам канального уровня DSL. Теперь в доме есть небольшая локальная сеть, реализованная на маршрутизаторе пользовательского класса, который обычно укомплектован коммутатором Ethernet и беспроводной точкой доступа.

Ключевая тема

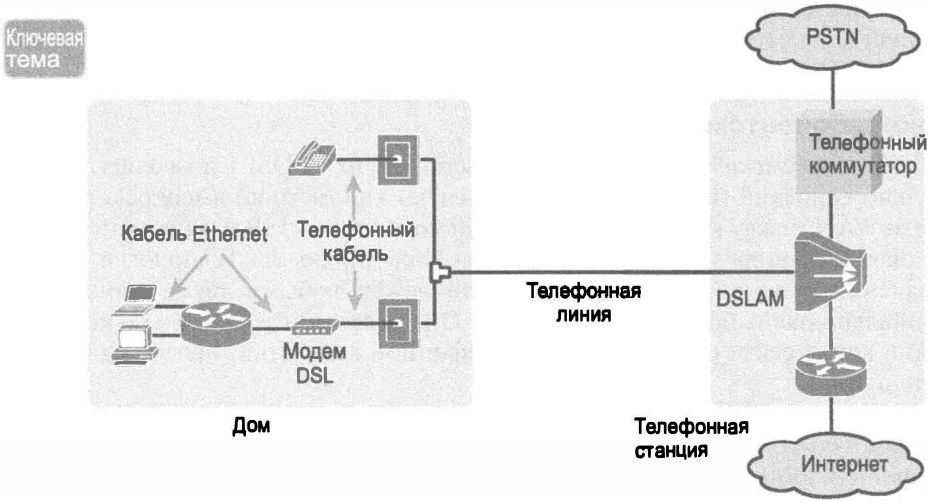


Рис. 3.16. Кабельная проводка и устройства DSL в доме

Домашний маршрутизатор должен быть в состоянии обмениваться данными с Интернетом. Для этого на телефонной станции используется *мультиплексор доступа DSL* (DSL Access Multiplexer — DSLAM), который отделяет и передает данные на маршрутизатор (см. рис. 3.16, *справа внизу*), обеспечивающий соединение с Интернетом. Устройство DSLAM отделяет также голосовые сигналы и передает их на телефонный коммутатор (см. рис. 3.16, *справа верху*).

Технология DSL обеспечивает телефонным компаниям удобную высокоскоростную службу доступа к Интернету, которую они могут предоставить своим клиентам. У телефонных компаний есть и другие возможности использования той же телефонной линии для данных, но они много сложнее DSL. Технология DSL поддер-

живает асимметричные скорости, т.е. скорость передачи от ISP к клиенту намного выше, чем обратно. Асимметричная скорость выгодней для потребительского доступа к Интернету из дома, поскольку щелчок на веб-странице посылает лишь несколько сот байтов в восходящий поток данных Интернета, но может вызвать передачу в ответ многих мегабайтов данных.

Кабельный Интернет

Служба доступа к Интернету по кабелю в общих чертах (но не в подробностях) очень похожа на DSL. Подобно DSL, кабельный Интернет использует для передачи данных уже существующую кабельную проводку (телевизионный кабель). Как и DSL, кабельный Интернет использует асимметричные скорости, получая данные быстрее, чем отправляя. Для большинства пользователей это лучше симметричной скорости. Подобно DSL, кабельный Интернет не предназначен для замены выделенных линий между любыми двумя площадками — он обеспечивает короткие каналы связи WAN между клиентом и ISP.

Кабельный Интернет использует те же базовые концепции кабельной проводки в доме, что и DSL. Рис. 3.17 имеет общее сходство с рис. 3.16, но устройства DSL заменены устройствами кабельного Интернета. Телефонная линия заменена коаксиальным кабелем провайдера кабельного телевидения, а модем DSL — кабельным модемом. В остальном домашняя проводка осталась той же.

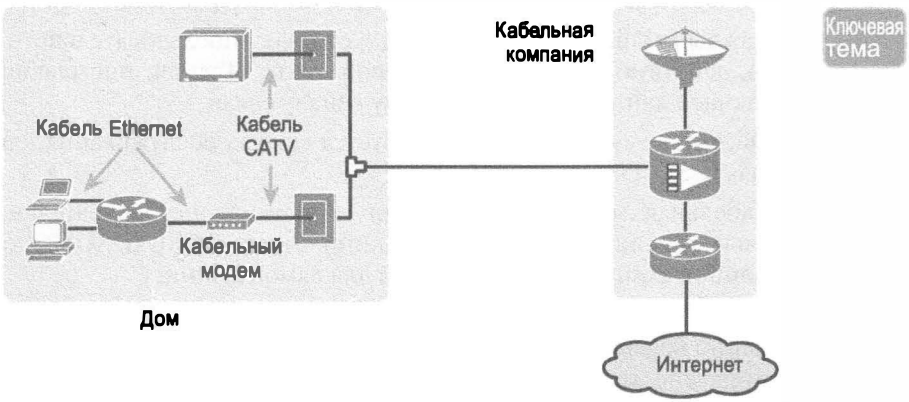


Рис. 3.17. Кабельная проводка и устройства в доме при кабельном Интернете

На своей стороне компания кабельной службы Интернета должна разделить данные и видео, как показано на рис. 3.17, *справа*. Данные уходят вниз, через маршрутизатор, а видеосигнал поступает сверху через устройства спутникового телевидения и передается на домашние телевизоры.

Кабельная служба Интернета и технология DSL непосредственно конкурируют за потребителя доступа к Интернету в малом бизнесе. По правде говоря, хотя обе технологии предлагают довольно высокие скорости, кабельный Интернет, как правило, быстрее DSL, но цены у провайдеров DSL немного ниже, что позволяет им конкурировать. Обе технологии поддерживают асимметричные скорости и постоянное подключение, обеспечивающее выход в Интернета без предварительных действий по установлению соединения.

Обзор

Резюме

- Поскольку и глобальные, и локальные сети соответствуют уровням 1 и 2 модели OSI, у них много сходств: обе определяют подробности кабельных соединений, скоростей передачи, кодировки, способов передачи данных по физическим каналам связи, а также логику передачи фреймов по каналу связи.
- Глобальные сети объединяют устройства, которые могут располагаться довольно далеко друг от друга, потенциально за сотни и даже тысячи километров.
- Для соединения локальных сетей применяют сеть WAN — объединенную сеть, использующую маршрутизаторы, подключенные к каждой из локальных сетей, и канал связи между ними.
- Термин *выделенная*, или *арендованная, линия* свидетельствует о том факте, что использующая выделенную линию компания не владеет каналом, а вносит ежемесячную арендную плату за его использование. Сейчас популярен термин *провайдер услуг*, обозначающий компанию, предоставляющую все формы подключения WAN, включая службы Интернета.
- На каждой площадке установлено клиентское оборудование (CPE), включающее маршрутизатор, плату последовательного интерфейса и модуль CSU/DSU.
  - Каждый маршрутизатор использует плату последовательного интерфейса, действующую подобно сетевой плате Ethernet, посылающей и получающей данные по физическому каналу связи.
  - Физическому каналу связи требуется модуль обслуживания канала и данных (CSU/DSU).
  - Кабельная проводка включает короткий последовательный кабель (только если используется внешний модуль CSU/DSU) и кабель телефонной компании, выделенный для самой линии.

Блок CSU/DSU обычно обеспечивает синхронизацию, указывая маршрутизатору, когда именно передавать по последовательному кабелю каждый бит. Последовательный интерфейс маршрутизатора способен обеспечивать синхронизацию, но маршрутизатор не делает этого без команды `clock rate`.

- Выделенная линия с протоколом HDLC создает канал связи WAN между двумя маршрутизаторами, чтобы они могли переправлять пакеты для устройств в локальных сетях на его концах. Сама выделенная линия представляет собой физические средства передачи битов в обоих направлениях. Фреймы HDLC — это средство инкапсуляции пакетов сетевого уровня при передаче по каналу связи между маршрутизаторами.
- Используемая в примерах сертификационных экзаменов CCENT и CCNA служба WAN Ethernet известна под двумя названиями: эмуляция Ethernet и Ethernet поверх MPLS (EoMPLS). Эмуляция Ethernet — это общий термин, означающий службу, действующую как один канал связи Ethernet. Служба EoMPLS относится к мультипротокольной меточной коммутации (MPLS) — одной из технологий, применяемой в облаке SP.

- Глобальные сети по самой своей природе предоставляют маршрутизаторам IP способ маршрутизации пакетов IP от локальной сети на одной площадке по сети WAN к другой локальной сети на другой площадке.
- Пользователи каналов связи Интернета зачастую используют такие технологии, как DSL и кабель. Эти технологии используют кабельную проводку, уже имеющуюся в большинстве домов, что существенно удешевляет подключение домашних пользователей. Технология DSL использует существующие аналоговые телефонные линии, а кабельный Интернет — телевизионные кабели.
  - Цифровой абонентский канал (DSL) позволяет создать относительно короткий (километры, но не десятки километров) высокоскоростной канал связи WAN между клиентом телефонной компании и ISP. Для него используется обычная домашняя телефонная линия из пары проводов.
  - Кабельный Интернет использует для передачи данных уже существующую кабельную проводку (телевизионный кабель).

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Какой из перечисленных ниже терминов наилучшим образом описывает основную функцию протоколов уровня 1 модели OSI применительно к WAN?

   А) Фреймирование (framing).

   Б) Пересылка последовательностей битов от одного устройства к другому.

   В) Адресация (addressing).

   Г) Обнаружение ошибок.

2. Какие типы устройств из перечисленных ниже обычно используют выделенные линии для подключения к четырехжильному телефонному проводу городской телефонной сети?

   А) Последовательный интерфейс маршрутизатора без внутреннего CSU/DSU.
   Б) Модуль CSU/DSU.
   В) Последовательный интерфейс маршрутизатора с внутренним трансивером.
   Г) Последовательный интерфейс коммутатора.

3. С какой скоростью может работать выделенная линия в Соединенных Штатах?

   А) 100 Мбит/с.

   Б) 100 Кбайт/с.

   В) 256 Кбайт/с.

   Г) 6.4 Мбит/с.

4. Какое из следующих полей в заголовке HDLC, используемом маршрутизаторами Cisco, было добавлено помимо стандарта ISO HDLC?

   А) Flag (Флаг).

   Б) Type (Тип).

   В) Address (Адрес).

   Г) FCS.

5. Два маршрутизатора, R1 и R2, подключены к службе MPLS по Ethernet. Она поддерживает двухточечную связь только между этими двумя маршрутизаторами, как службу Ethernet второго уровня. Что из приведенного ниже вероятнее всего справедливо для этой WAN? (Выберите два ответа.)

   А) Маршрутизатор R1 подключен к физическому каналу связи Ethernet, к другому концу которого подключен маршрутизатор R2.

   Б) Маршрутизатор R1 подключен к физическому каналу связи Ethernet, к другому концу которого подключено устройство в представительстве провайдера услуг WAN.

   В) Маршрутизатор R1 перенаправит фреймы канала связи на маршрутизатор R2, используя заголовок и концевик протокола HDLC.

   Г) Маршрутизатор R1 перенаправит фреймы канала связи на маршрутизатор R2, используя заголовок и концевик протокола Ethernet.

6. Какие из следующих технологий доступа к Интернету, используемых для соединения площадки и ISP, предоставляют асимметричные скорости передачи? (Выберите два ответа).

   А) Выделенные линии.

   Б) DSL.

   В) Кабельный Интернет.

   Г) BGP.

7. Фред только что установил дома службу DSL с отдельным модемом DSL и маршрутизатором потребительского класса с четырьмя портами Ethernet. Фред хочет использовать тот же старый телефон, что и до установки DSL. Что произойдет с телефоном и кабельной проводкой при установке подключения DSL?

   А) Используется старый телефон, подключенный к одному из портов Ethernet маршрутизатора или коммутатора.

   Б) Используется старый телефон, подключенный к одному из портов модема DSL.

   В) Используется старый телефон, подключенный к одной из существующих розеток, а не к новому устройству.

   Г) Старый телефон следует заменить цифровым.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы приведены в табл. 3.3.

Таблица 3.3. Ключевые темы главы 3

Элемент	Описание	Страница
Рис. 3.4	Типичная схема кабельной проводки CPE для выделенной линии	117
Рис. 3.9	Оптоволоконный канал связи Ethernet, соединяющий маршрутизатор CPE с глобальной сетью провайдера услуг	122
Рис. 3.14	Три примера каналов связи Интернета	127
Рис. 3.16	Кабельная проводка и устройства DSL в доме	128
Рис. 3.17	Кабельная проводка и устройства в доме при кабельном Интернете	129

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

выделенная линия (leased line), распределенная сеть (Wide-Area Network — WAN), телефонная компания (telco), последовательный интерфейс (serial interface), HDLC, DSL, кабельный Интернет (cable Internet), модем DSL (DSL modem), Ethernet поверх MPLS (Ethernet over MPLS)

Ответы на контрольные вопросы:
1 Б. **2** Б. **3** В. **4** Б. **5** Б и Г. **6** Б и В. **7** В.

ГЛАВА 4

Основы IPv4-адресации и маршрутизации

Сетевой уровень модели TCP/IP (уровень 3) определяет правила доставки пакетов IP от первоначального устройства, создавшего пакет, на устройство его получателя. Этот процесс требует взаимодействия нескольких разных устройств и применения ряда концепций.

Настоящая глава начинается с краткого обзора всех взаимодействующих функций, а затем они будут рассмотрены более подробно.

Маршрутизация IP. Процесс перенаправления пакетов IP хостам и маршрутизаторам (уровень 3 PDU) по локальным и глобальным сетям.

IP-адресация. Адреса, идентифицирующие хосты отправителя и получателя пакета. Правила адресации организуют адреса в группы, что весьма помогает процессу маршрутизации.

Протокол маршрутизации IP. Протокол, позволяющий маршрутизаторам динамически узнавать о группах IP-адресов, чтобы маршрутизатор знал, куда перенаправлять пакеты IP, чтобы они достигли хоста назначения.

Другие утилиты. Сетевой уровень полагается также и на другие утилиты. Для протокола TCP/IP это система доменных имен (DNS), протокол преобразования адресов (ARP) и утилита ping.

Обратите внимание, что у всех этих функций есть варианты для протокола IP версии 4 (IPv4) и для более новой версии 6 (IPv6). Эта глава посвящена протоколу IPv4 и всему связанному с ним. Те же функции для протокола IPv6 рассматриваются в части VII.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы.

Выбор компонентов сети, удовлетворяющих заданной спецификации.

Передача данных между двумя хостами по сети.

Технологии маршрутизации IP

Различия методов маршрутизации и протоколов маршрутизации:

Статика или динамика.

Основные темы

Обзор функций сетевого уровня

Хотя за последние годы использовалось много моделей протоколов, ныне доминирует модель TCP/IP. На сетевом уровне модели TCP/IP также есть две разновидности основного протокола, вокруг которого строятся все другие функции сетевого уровня: протокол IP версии 4 (IPv4) и протокол IP версии 6 (IPv6). Обе версии протокола (IPv4 и IPv6) определяют одинаковые функции сетевого уровня, но с разными деталями. В этой главе рассматриваются функции сетевого уровня протокола IPv4, а о протоколе IPv6 речь пойдет в части VII.

ВНИМАНИЕ!

Все упоминания протокола IP в этой главе относятся к версии IPv4.

Протокол IP отвечает за маршрутизацию данных в форме пакетов IP от хоста отправителя к хосту получателя. Он не участвует в физической передаче данных, — это задача более низких уровней модели TCP/IP. Протокол IP определяет логические детали передачи данных, а не физические. В частности, сетевой уровень определяет правила следования пакетов по сети TCP/IP, даже когда пакет пересекает множество каналов связи WAN и LAN разных типов.

Сначала обсудим сетевой уровень модели TCP/IP и рассмотрим адресацию и маршрутизацию IP. Эти две темы взаимосвязаны, поскольку маршрутизация IP полагается на структуру и значение IP-адресов, а IP-адресация была разработана с учетом маршрутизации IP. Далее следует краткий обзор протоколов маршрутизации, позволяющих маршрутизаторам изучать информацию, необходимую для правильного перенаправления пакетов.

Логика маршрутизации (перенаправления) сетевого уровня

Для осуществления маршрутизации IP маршрутизаторы и компьютеры конечного пользователя (*хосты* (host) сети TCP/IP) должны взаимодействовать. В операционной системе хоста выполняется программное обеспечение, реализующее сетевой уровень модели TCP/IP. Хосты используют это программное обеспечение для выбора соседнего маршрутизатора, на который следует послать пакеты IP. Этот маршрутизатор в свою очередь выбирает направление дальнейшей передачи пакета IP. Совместно хосты и маршрутизаторы доставляют пакет IP конечному получателю, как показано на рис. 4.1.

Созданный компьютером PC1 пакет IP в верхней части рисунка следует к компьютеру PC2 внизу. Давайте на этом примере обсудим логику маршрутизации сетевого уровня, используемую каждым устройством вдоль этого пути.

ВНИМАНИЕ!

Термин *выбор пути* (path selection) иногда используется как синоним *процесса маршрутизации* (routing process), представленного на рис. 4.1. В других случаях он относится к протоколам маршрутизации, а именно к выбору протоколами маршрутизации наилучшего маршрута из всех возможных к тому же получателю.

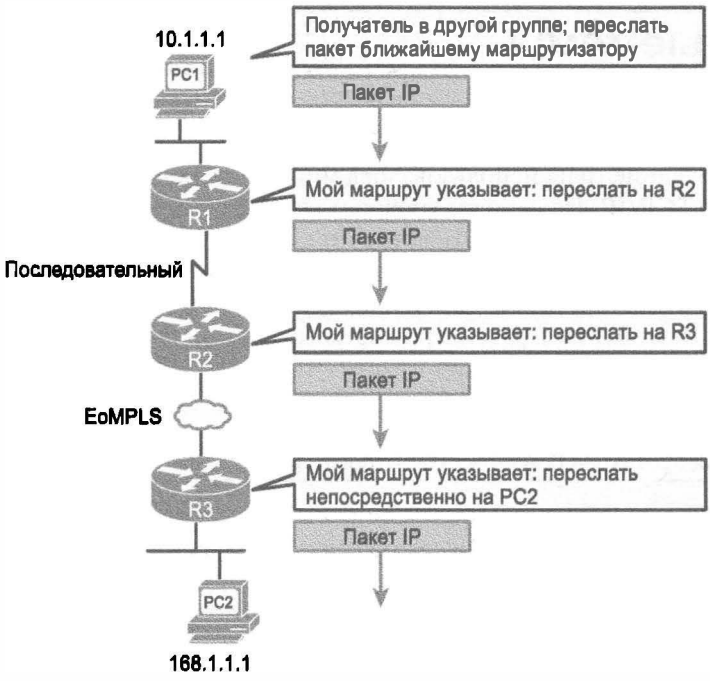


Рис. 4.1. Логика маршрутизации: компьютер PC1 посылает пакет IP на компьютер PC2

Логика хоста: передача пакета на стандартный маршрутизатор

В этом примере после несложного анализа компьютер PC1 решает послать пакет IP маршругизатору, чтобы он перенаправил его дальше. Компьютер PC1 анализирует адрес получателя (168.1.1.1) и понимает, что компьютер PC2 находится не в той же локальной сети, что и он сам. Таким образом, логика хоста PC1 требует послать пакет на устройство, которое должно знать, куда перенаправить данные, — соседний маршрутизатор в той же локальной сети, называемый стандартным маршрутизатором.

Чтобы послать пакет IP на стандартный маршрутизатор, отправитель посылает фрейм канала связи через передающую среду на соседний маршрутизатор; этот фрейм содержит пакет в части данных. Для гарантии получения фрейма соседним маршрутизатором используется адрес канального уровня (уровня 2), находящийся в заголовке канала связи.

ВНИМАНИЕ!

Стандартный маршрутизатор (default router) называется также *стандартным шлюзом* (default gateway).

Логика маршрутизаторов R1 и R2: перенаправление данных в сети

Все маршрутизаторы используют одинаковый процесс перенаправления пакета. Каждый маршрутизатор хранит *таблицу маршрутизации IP* (IP routing table). Эта таблица содержит *группировки* (grouping) IP-адресов, известные также как *сети IP*

(IP network) и *подсети IP* (IP subnet). Когда маршрутизатор получает пакет, он сравнивает IP-адрес получателя пакета с записями в таблице маршрутизации и находит соответствие. Найденная запись содержит список направлений, указывающих маршрутизатору, куда перенаправить пакет далее.

Маршрутизатор R1 на рис. 4.1 нашел бы в таблице маршрутизации запись с адресом получателя 168.1.1.1, которая указала бы ему передать пакет далее на маршрутизатор R2. Аналогично маршрутизатор R2 нашел бы в таблице маршрутизации запись, указывающую переслать пакет далее по каналу связи EoMPLS на маршрутизатор R3.

Концепция маршрутизации напоминает движение по автостраде с перекрестками, указатели над которыми указывают на ближайшие города. Это позволяет выбрать путь к каждому городу. Точно так же маршрутизатор просматривает таблицу маршрутизации IP (эквивалент дорожных указателей) и направляет каждый пакет по пути к следующей локальной сети или каналу связи WAN (эквивалент дорог).

Логика маршрутизатора R3: доставка данных конечному получателю

Последний маршрутизатор в пути, R3, использует почти ту же логику, что и R1 и R2, но с одним незначительным отличием. Маршрутизатор R3 должен перенаправить пакет непосредственно на компьютер PC2, а не на некий другой маршрутизатор. На первый взгляд отличие кажется незначительным. Но когда в следующем разделе вы узнаете о том, как сетевой уровень использует локальные и глобальные сети, это отличие станет очевидным.

Как маршрутизация сетевого уровня использует локальные и глобальные сети

Хотя логика сетевого уровня маршрутизации игнорирует физические детали передачи, биты все же должны быть переданы. Для этого логика сетевого уровня на хосте или маршрутизаторе должна сдать пакет протоколу канального уровня, который, в свою очередь, задействует физический уровень для фактической передачи данных. Канальный уровень добавляет к пакету соответствующий заголовок и концевик, формируя отправляемый по физической сети фрейм (см. главу 2).

Процесс маршрутизации перенаправляет пакет сетевого уровня из конца в конец сети, а каждый фрейм канала связи — только на своем участке сети. Каждый последующий фрейм канального уровня переносит пакет на следующее устройство, которое в свою очередь применяет логику сетевого уровня. Короче говоря, сетевой уровень решает общую задачу: как переслать этот пакет на следующее заданное устройство, а канальный уровень заботится о специфических особенностях: как инкапсулировать этот пакет во фрейме канала связи и передать его. Рис. 4.2 демонстрирует основы логики инкапсуляции на каждом устройстве, используя тот же пример, что и на рис. 4.1.

Поскольку маршрутизаторы создают новые заголовки и концевики канала связи, а эти новые заголовки содержат адреса канала связи, у компьютеров и маршрутизаторов должен быть некий способ решать, как использовать адреса канала связи. Примером механизма определения используемого маршрутизатором адреса канала связи является *протокол преобразования адресов* (Address Resolution Protocol — ARP). *Он динамически изучает адрес канала связи хоста IP, подключенного к локальной сети.* Например, на последнем этапе (рис. 4.2, *снизу*) маршрутизатор R3 использовал бы протокол ARP для выяснения MAC-адреса компьютера PC2, прежде чем посылать на него любые пакеты.

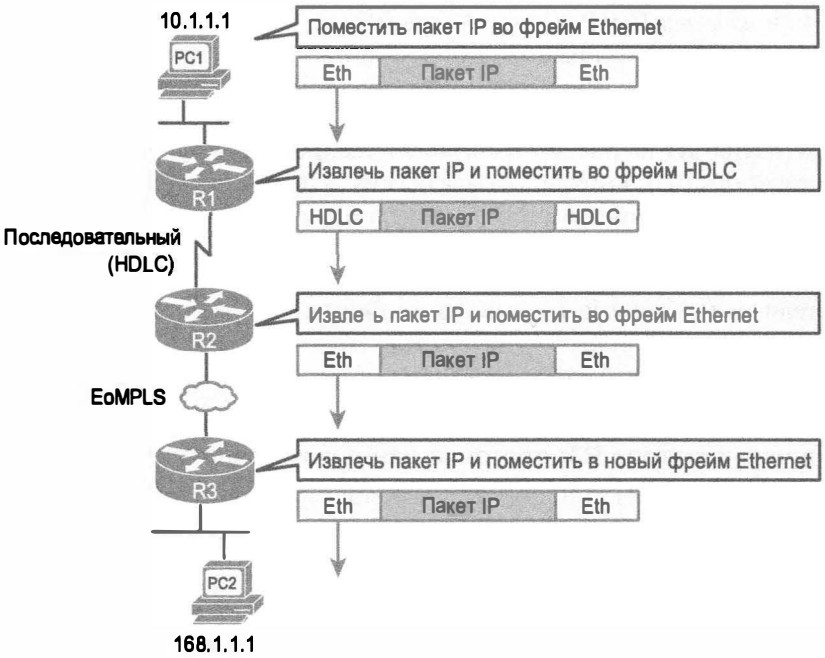


Рис. 4.2. Инкапсуляция сетевого и канального уровней

Маршрутизация, как уже было сказано, в своей основе имеет две главные идеи.

- Процесс маршрутизации перенаправляет пакеты третьего уровня, которые также называются *блоками данных протокола уровня 3* (Layer 3 Protocol Data Units — L3 PDU), на основе содержащегося в пакете адреса получателя.
- Процесс маршрутизации использует канальный уровень для инкапсуляции пакетов третьего уровня во фреймы второго уровня для передачи через каждый последующий канал.

IP-адресация и как она помогает маршрутизации IP

Протокол IP определяет адреса сетевого уровня, идентифицирующие любой хост или интерфейс маршрутизатора, подключенный к сети TCP/IP. Идея подобна почтовому адресу: любой интерфейс, собирающийся получать пакеты IP, должен иметь IP-адрес, как любой собирающийся получать письма по почте должен иметь почтовый адрес.

Протокол TCP/IP группирует IP-адреса так, чтобы адреса, используемые в той же физической сети, принадлежали той же группе. Такие группы адресов известны как *сеть IP* (IP network) или *подсеть IP* (IP subnet). Используя прежнюю аналогию с почтовой службой, каждая сеть и подсеть IP работают как почтовый индекс. Все соседние почтовые адреса имеют одинаковый индекс, а все соседние IP-адреса — одинаковый номер сети или подсети IP.

ВНИМАНИЕ!

В терминологии IP слово *сеть* означает очень много разных понятий. Чтобы избежать разночтений, в контексте IP-адресации в этой книге (и других) не используют термин *сеть* для других концепций. В частности, для обозначения сети, состоящей из маршрутизаторов, коммутаторов, кабелей и другого оборудования, в этой книге используется термин *объединенная сеть*.

Протокол IP определяет правила, согласно которым IP-адрес может относиться к той же сети или подсети IP. Числовое представление адресов в той же группе имеет одинаковое значение в первой части адресов. Для сетей на рис. 4.1 и 4.2 возможны следующие соглашения:

- хосты в верхней сети Ethernet: адреса начинаются с 10;
- хосты на последовательном канале R1-R2: адреса начинаются с 168.10;
- хосты в канале связи EoMPLS R2-R3: адреса начинаются с 168.11;
- хосты в нижней сети Ethernet: адреса начинаются с 168.1.

Системе почтового сообщения требуется, чтобы местные органы власти присваивали адреса всем новым зданиям. Было бы смешно, если бы два дома, стоящих рядом, имели адреса с разными почтовыми индексами. Точно так же было бы глупо иметь адреса с одинаковым почтовым индексом на противоположных концах страны.

Аналогично, чтобы сделать маршрутизацию более эффективной, протоколы сетевого уровня группируют адреса по их положению и значению адреса. Так маршрутизатор сможет хранить в таблице маршрутизации только одну запись для каждой сети или подсети IP, а не по отдельной записи для каждого конкретного IP-адреса.

Процесс маршрутизации использует также заголовок IPv4 (рис. 4.3), который включает 32-разрядный IP-адрес отправителя и 32-разрядный IP-адрес получателя. В заголовке есть также другие поля, и некоторые из них обсуждаются в других частях этой книги, а пока вам достаточно знать о наличии в заголовке IP 20-байтовых полей IP-адресов отправителя и получателя.

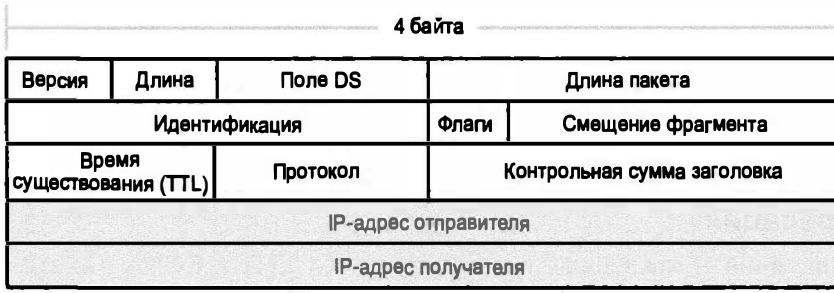


Рис. 4.3. Заголовок IPv4

Протоколы маршрутизации

Чтобы логика маршрутизации работала на хостах и маршрутизаторах, каждый из них должен иметь информацию об объединенной сети TCP/IP. Чтобы хосты могли посылать пакеты дистанционным получателям, они должны знать IP-адрес своего стандартного маршрутизатора, а чтобы маршрутизаторы могли перенаправлять пакеты соответствующей сети или подсети IP, они должны знать маршруты к ним.

Хотя сетевой инженер вполне может вручную настроить (ввести) все необходимые маршруты на каждом маршрутизаторе, большинство из них позволяет выполнить эту работу протоколу маршрутизации. Если запустить тот же протокол маршрутизации на всех маршрутизаторах в объединенной сети TCP/IP (с правильными параметрами), то маршрутизаторы начнут между собой обмен сообщениями протокола маршрутизации. В результате все маршрутизаторы изучат маршруты для всех сетей и подсетей IP в объединенной сети TCP/IP.

На рис. 4.4 приведен пример использования той же схемы, что и на рис. 4.1 и 4.2. В данном случае сеть IP 168.1.0.0, состоящая из всех адресов, начинающихся с 168.1, находится в канале Ethernet (см. рис. 4.4, *снизу*). Зная этот факт, маршрутизатор R3 посылает сообщение протокола маршрутизации на маршрутизатор R2 (этап 1). Маршрутизатор R2 изучает маршрут для сети 168.1.0.0 (см. рис. 4.4, *слева*). На этапе 2 маршрутизатор R2 изменяет и посылает сообщение протокола маршрутизации на маршрутизатор R1, чтобы у него теперь был маршрут к сети IP (168.1.0.0).

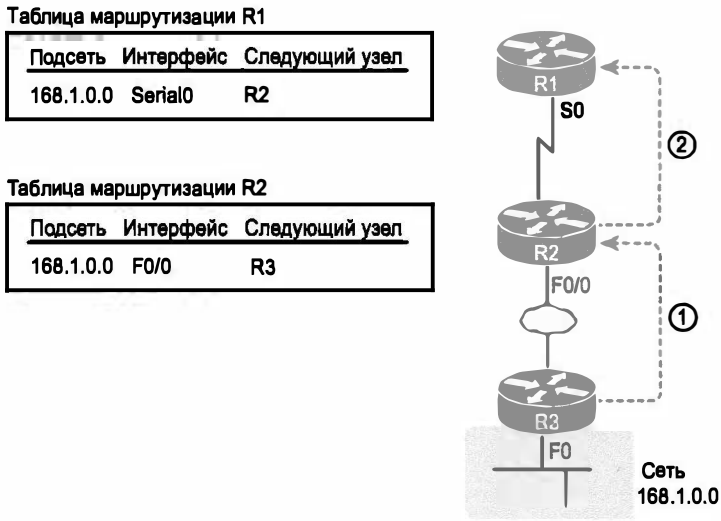


Рис. 4.4. Анонсы протоколов маршрутизации о сетях и подсетях

На этом завершается краткий обзор работы сетевого уровня модели TCP/IP. Оставшаяся часть главы посвящена детальному рассмотрению ее ключевых компонентов.

IPv4-адресация

IPv4-адресация — важнейшая тема экзаменов CCENT и CCNA. По завершении изучения этой книги вы должны уверенно разбираться в IP-адресах, их форматах, группировке, правилах объединения в подсети, интерпретации документации существующих сетей IP и т.д. Проще говоря, адресацию и подсети следует знать в совершенстве!

В этом разделе содержится введение в IP-адресацию и подсети, а также рассматриваются концепции, лежащие в основе структуры IP-адресов, включая ее отношение к маршрутизации IP. В частях III и V представлены подробности концепций и математических механизмов, лежащих в основе IPv4-адресации и создания подсетей.

Правила IP-адресов

Если устройство должно общаться по протоколу TCP/IP, то ему нужен IP-адрес. Когда у устройства есть IP-адрес, его программное обеспечение и оборудование могут посылать и получать пакеты IP. Любое устройство, обладающее по крайней мере одним интерфейсом с IP-адресом и способное обмениваться пакетами IP, называется *хостом IP* (IP host).

IP-адреса состоят из 32-разрядного числа, обычно записанного в *десятичном представлении с разделительными точками* (Dotted-Decimal Notation — DDN). Термин "десятичное" свидетельствует о том, что каждый байт (8 битов) 32-разрядного IP-адреса представляется как его десятичный эквивалент. Четыре получающихся десятичных числа разделены точками, отсюда "с разделительными точками". Рассмотрим IP-адрес 168.1.1.1, записанный в десятичном представлении с разделительными точками; его фактическая, двоичная, версия такова: 10101000 00000001 00000001 00000001. (Двоичную форму записи почти никогда не используют, но в приложении А есть таблица, позволяющая легко преобразовать двоичное представление в DDN, и наоборот.)

У каждого IP-адреса в представлении DDN есть четыре десятичных *октета* (octet), разделенных точками. Термин *октет* — это синоним термина *байт*. Поскольку каждый октет представляет 8-битовое двоичное число, его диапазон десятичных чисел составляет 0–255 включительно. Например, первый октет IP-адреса 168.1.1.1 — это 168, второй — 1 и т.д.

И наконец, обратите внимание на то, что каждый сетевой интерфейс использует уникальный IP-адрес. Большинство людей полагают, что IP-адрес есть у их компьютера, но фактически он принадлежит сетевой плате компьютера. Например, если на портативном компьютере одновременно будут работать и плата сетевого интерфейса Ethernet (NIC), и беспроводная сетевая плата, то у обеих будет свой IP-адрес. Точно так же у маршрутизаторов, обладающих обычно несколькими сетевыми интерфейсами, которые перенаправляют пакеты IP, IP-адрес есть у каждого интерфейса.

Правила группировки IP-адресов

В первоначальных спецификациях стека протоколов TCP/IP IP-адреса группировались в наборы последовательных адресов, которые назывались *сетями IP* (IP network). Все адреса в одной сети IP имели одинаковое значение первой части. На рис. 4.5 показана простая объединенная сеть, состоящая из трех отдельных сетей IP.

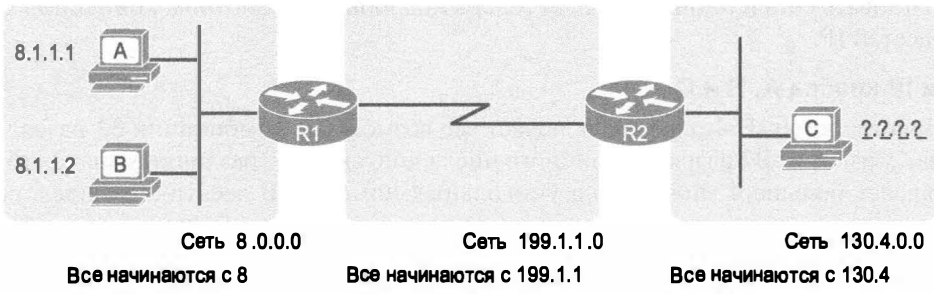


Рис. 4.5. Пример объединенной сети TCP/IP, использующей номера сети IPv4

На рисунке представлены *идентификаторы сети* (network ID) для каждой сети, а также их десятичные значения. Например, хосты в локальной сети Ethernet крайней левой сети используют IP-адреса, начинающиеся с первого октета 8; идентификатором этой сети является 8.0.0.0. Последовательный канал связи между маршрутизаторами R1 и R2 состоит только из двух последовательных интерфейсов (по одному на каждом маршрутизаторе) и использует IP-адрес, начинающийся с трех октетов 199.1.1.

Рис. 4.5 является также хорошей иллюстрацией двух важных фактов о группировке IPv4-адресов.

Ключевая тема **Правила группировки IP-адресов в сети или подсети**

- IP-адреса в одной группе не должны отделяться друг от друга маршрутизатором;
- IP-адреса, разделенные маршрутизатором, должны находиться в разных группах.

Согласно первому правилу, хосты A и B слева находятся в той же сети IP и имеют IP-адреса, начинающиеся с 8. Хосты A и B не могут быть отделены друг от друга маршрутизатором (и так оно и есть).

Согласно второму правилу, хост C, отделенный от хоста A по крайней мере одним маршрутизатором, не может быть в той же сети IP, что и хост A. Адрес хоста C не может начаться с 8.

ВНИМАНИЕ!

В этом примере использовались только сети IP и никаких подсетей только потому, что подсети еще не обсуждались.

Как уже упоминалось, группировка IP-адресов подобна почтовым индексам. Все адреса с почтовым индексом автора расположены в небольшом городе штата Огайо. Если бы некоторые из адресов с тем же индексом были в Калифорнии, то почта иногда доставлялась бы не тому почтовому отделению, поскольку почтовая служба доставляет письма на основании почтового индекса. Почтовая система полагается на то, что все адреса с одинаковым индексом находятся рядом друг с другом.

Аналогично маршрутизация IP полагается на то, что все адреса в одной сети или подсети IP расположены в той же области, а именно на том же канале связи WAN или LAN. В противном случае маршрутизаторы могли бы доставить пакеты IP не тем областям.

Для любой объединенной сети TCP/IP каждый канал связи LAN и WAN будет использовать сеть или подсеть IP. А теперь подробнее рассмотрим концепции сетей и подсетей IP.

Сети IP класса A, B и C

Пространство IPv4-адресов включает все возможные комбинации 32-разрядного числа. А именно: 32 разряда двоичного числа допускает 2^{32} различных значений, что составляет больше 4 миллиардов уникальных номеров. В десятичном представлении с разделительными точками эти числа включают все комбинации значений от 0 до 255 во всех четырех октетах: 0.0.0.0, 0.0.0.1, 0.0.0.2 и так далее до 255.255.255.255.

Стандарт IP разделяет все пространство адресов на классы, идентифицируемые по значению первого октета. Класс A включает примерно половину пространства IPv4-адресов с номерами, начинающимися на 1–126, как показано на рис. 4.6. Класс B включает четверть всего пространства адресов с номерами, начинающимися на 128–191, а класс C включает одну восьмую пространства адресов с номерами, начинающимися на 192–223.

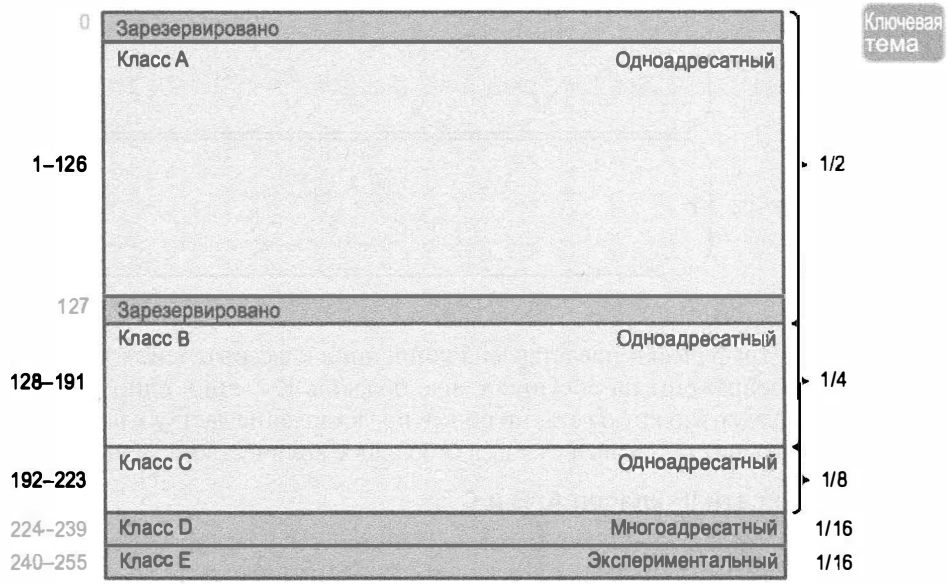


Рис. 4.6. Распределение пространства IPv4-адресов по классам

На рис. 4.6 показана также цель этих пяти классов адресов. Классы A, B и C определяют одноадресатные IP-адреса, т.е. адреса, идентифицирующие один интерфейс хоста. Класс D определяет многоадресатные адреса, используемые для передачи одного пакета нескольким хостам. Их номера начинаются с 224–239. Класс E определяет экспериментальные адреса с номерами, начинающимися на 240–255.

Стандарты IPv4 разделяют также одноадресатные классы A, B и C на предопределенные сети IP. Каждая сеть IP представляет собой подмножество значений адресов в классе.

Протокол IPv4 использует три класса одноадресатных адресов для того, чтобы сети IP в каждом классе могли иметь разный размер и применяться для разных задач. Сети класса A предназначены для очень большого количества IP-адресов (больше 16 миллионов адресов хоста на сеть IP). Но поскольку каждая сеть класса A настолько велика, к нему относится всего 126 сетей. Класс B определяет сети IP, насчитывающие по 65 534 адреса на сеть, но самих таких сетей более 16 000. Класс C определяет намного меньшие сети IP, по 254 адреса в каждой, как показано на рис. 4.7.

Ключевая тема

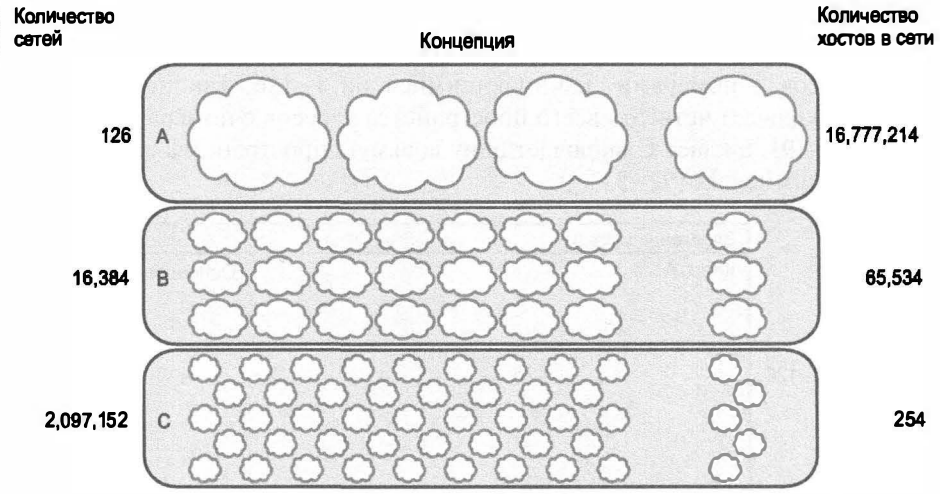


Рис. 4.7. Размер частей сети и хоста у адресов сетей классов A, B и C

На рис. 4.7 графически представлена концепция классов сетей, а также номера сети. Сети IP изображены на рисунке в виде облаков. Конечно, одно облако не представляет каждую возможную сеть, но общее представление дает: у класса A небольшое количество больших облаков, а у класса C большое количество маленьких облаков.

Фактические сети IP класса A, B и C

На рис. 4.7 представлено общее количество сетей IP классов A, B и C во всем мире. В конечном счете, чтобы создать рабочую объединенную сеть TCP/IP, фактически необходимо выбрать и использовать некую из этих сетей IP. Для этого необходимо быть в состоянии ответить на вопрос: как идентифицировать конкретную сеть IP?

В первую очередь нужна возможность идентифицировать каждую сеть. Для этого используется *идентификатор сети* (network ID). Идентификатор сети — это только одно из зарезервированных значений адреса в сети, которое идентифицирует саму сеть IP. (Идентификатор сети не может использоваться как IP-адрес хоста). Например, в табл. 4.1 приведены идентификаторы сети, соответствующие сетям на рис. 4.5.

Таблица 4.1. Идентификаторы сети, использованные на рис. 4.5

Концепция	Класс	Идентификатор сети
Все адреса начинаются с 8	A	8.0.0.0
Все адреса начинаются с 130.4	B	130.4.0.0
Все адреса начинаются с 199.1.1	C	199.1.1.0

ВНИМАНИЕ!

Одни используют термин *идентификатор сети* (network ID), другие — *номер сети* (network number), третьи — *адрес сети* (network address). Все эти три термина — синонимы.

Так каковы же фактические идентификаторы сетей IP классов A, B и C? Сначала рассмотрим сети класса A. Согласно рис. 4.7, существуют только 126 сетей класса A. Они состоят из всех адресов, начинающихся с 1, всех адресов, начинающихся с 2,

всех адресов, начинающихся с 3, и так далее до 126. Список некоторых из сетей класса A приведен в табл. 4.2.

Таблица 4.2. Примеры сетей IPv4 класса A

Концепция	Класс	Идентификатор сети
Все адреса начинаются с 8	A	8.0.0.0
Все адреса начинаются с 13	A	13.0.0.0
Все адреса начинаются с 24	A	24.0.0.0
Все адреса начинаются с 125	A	125.0.0.0
Все адреса начинаются с 126	A	126.0.0.0

У сетей класса B значение первого октета находится в диапазоне от 128 до 191, но у адресов сетей класса B одинаковое значение уже двух первых октетов. Например, сеть класса B на рис. 4.5 имеет идентификатор 130.4.0.0. Значение 130.4.0.0 принадлежит классу B, поскольку его первый октет принадлежит диапазону 128–191, а адрес конкретной сети класса определяют первые два октета. Список некоторых из сетей класса B приведен в табл. 4.3.

Таблица 4.3. Примеры сетей IPv4 класса B

Концепция	Класс	Идентификатор сети
Все адреса начинаются с 128.1	B	128.1.0.0
Все адреса начинаются с 172.20	B	172.20.0.0
Все адреса начинаются с 191.191	B	191.191.0.0
Все адреса начинаются с 150.1	B	150.1.0.0

Сети класса C также могут быть легко идентифицированы по принадлежности первого значения октета диапазону от 192 до 223 включительно. У адресов сетей класса C одинаковы первые три октета — они определяют группу адресов единой сети класса C. Список некоторых из сетей класса C приведен в табл. 4.4.

Таблица 4.4. Примеры сетей IPv4 класса C

Концепция	Класс	Идентификатор сети
Все адреса начинаются с 199.1.1	C	199.1.1.0
Все адреса начинаются с 200.1.200	C	200.1.200.0
Все адреса начинаются с 223.1.10	C	223.1.10.0
Все адреса начинаются с 209.209.1	C	209.209.1.0

Список всех сетей классов A, B и C, конечно, занял бы намного больше места. Сугубо в демонстрационных целях в табл. 4.5 приведен диапазон значений первого октета, идентифицирующего класс, и общий диапазон IPv4-адресов сети классов A, B и C.

ВНИМАНИЕ!

Термин *классовая сеть IP* (classful IP network) обозначает любую сеть класса A, B или C, поскольку они определяются правилами для классов A, B и C.

Ключевая тема

Таблица 4.5. Все возможные корректные адреса сетей

Класс	Диапазон первого октета	Допустимые адреса сетей
A	От 1 до 126	От 1.0.0.0 до 126.0.0.0
B	От 128 до 191	От 128.0.0.0 до 191.255.0.0
C	От 192 до 223	От 192.0.0.0 до 223.255.255.0

Создание подсетей IP

Создание подсетей — одна из важнейших тем сертификационных экзаменов CCENT и CCNA. Чтобы решать задачи подсетей в реальной жизни и на экзамене, необходимо твердо знать, как они работают и как осуществлять необходимые вычисления. Части III и V этой книги посвящены подробностям концепций создания подсетей и их математическим механизмам, однако, прежде чем приступать к этим темам, имеет смысл получить о них общее представление.

Создание подсетей — это дальнейшее разделение пространства IPv4-адресов на группы размером меньше одной сети IP. Подсети IP позволяют взять одну сеть IP класса A, B или C и разделить ее на множество меньших групп последовательных IP-адресов. Фактически термин subnet (*подсеть*) — это сокращение от subdivided network (*разделенная сеть*). После разделения в каждой области, где обычно использовалась бы вся сеть класса A, B или C, можно использовать меньшую подсеть, тратя впустую меньше IP-адресов.

Давайте проясним, как объединенная сеть может одновременно использовать классовые сети IPv4 и подсети классовых сетей IPv4. На рисунках ниже показана та же объединенная сеть, но на первом рисунке только с классовыми сетями, и только с подсетями на втором. Рис. 4.8 демонстрирует использование пяти сетей класса B без подсетей.

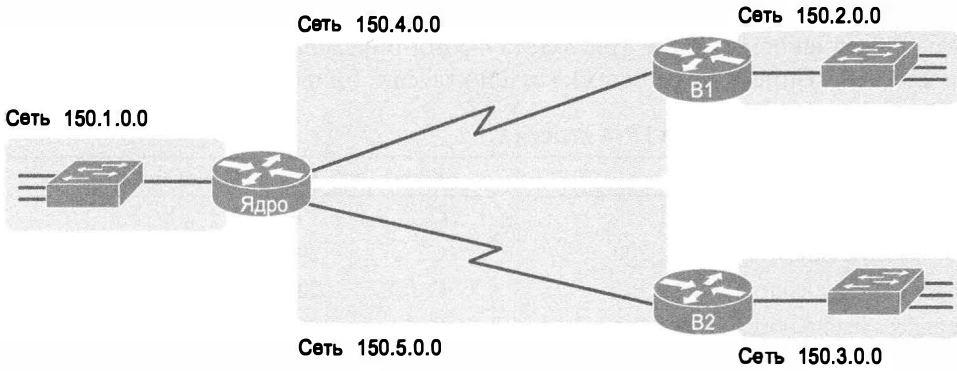


Рис. 4.8. Пример использования пяти сетей класса B

Проекту на рис. 4.8 требуется пять групп IP-адресов, каждая из которых является в данном случае сетью класса B, а именно: три локальных сети и два последовательных канала связи, для всех их используются сети класса B.

Это тратит впустую множество IP-адресов, поскольку каждая сеть класса B насчитывает $2^{16} - 2$ адреса хоста, что значительно больше, чем может когда-либо по-

надобиться для локальной сети или канала связи WAN. Например, сеть Ethernet слева использует все 65 534 IP-адреса сети класса B, начинающихся с 150.1. Однако одна локальная сеть редко насчитывает даже несколько сотен устройств, поэтому большинство IP-адресов сети класса B 150.1.0.0 было бы потрачено впустую. Еще большие растраты происходят на двухточечных последовательных каналах связи, нуждающихся только в двух IP-адресах.

На рис. 4.9 представлен более современный проект, использующий простые подсети. Как и на предыдущем рисунке, этот проект нуждается в пяти группах адресов. Однако в данном случае используется пять подсетей сети класса B 150.9.0.0.

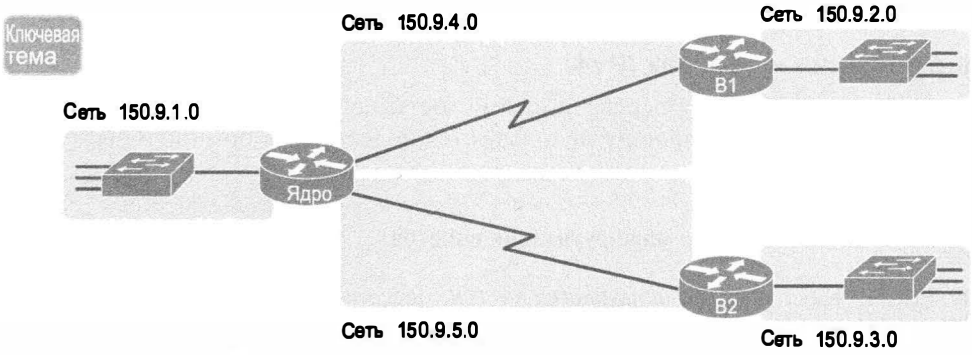


Рис. 4.9. Концептуальное представление подсетей для проекта на рис. 4.8

Создание подсетей позволяет сетевому инженеру выбирать для объединенной сети TCP/IP более или менее длинную часть адреса, которая будет совпадать у всех ее адресов. Это обеспечивает немного гибкости, но на рис. 4.9 приведена одна из самых простых форм подсетей. В данном случае каждая подсеть включает адреса, начинающиеся с одинакового значения первых трех октетов:

- группа из 254 адресов, начиная с 150.9.1;
- группа из 254 адресов, начиная с 150.9.2;
- группа из 254 адресов, начиная с 150.9.3;
- группа из 254 адресов, начиная с 150.9.4;
- группа из 254 адресов, начиная с 150.9.5.

В результате использования подсетей сетевой инженер сэкономил много IP-адресов, ведь использована была лишь небольшая часть сети 150.9.0.0 класса B. У каждой подсети 254 адреса, которых вполне должно хватить для каждой локальной сети и более чем достаточно для каналов связи WAN.

ВНИМАНИЕ!

В частях III и V этой книги рассматриваются подробности IP-адресации, включая методы выбора сети IP и ее разделения на меньшие подсети.

Ознакомившись с некоторыми из деталей IP-адресации, обратим внимание на то, какое отношение они имеют к маршрутизации. У каждого хоста и интерфейса маршрутизатора есть IP-адрес. Однако IP-адреса выбраны не беспорядочно, они

сгруппированы, чтобы помочь процессу маршрутизации. Группы адресов могут быть либо всем классом A, B или C, либо быть подсетью.

Маршрутизация IPv4

В первом разделе этой главы основы маршрутизации IPv4 рассматривались на примере сети с тремя маршрутизаторами и двумя компьютерами. Теперь, обладая некоторыми знаниями об IP-адресации, можно подробнее рассмотреть процесс маршрутизации IP. Этот раздел начинается с примера простой логики маршрутизации из двух этапов на передающем хосте, а затем перейдем к обсуждению того, как маршрутизаторы выбирают маршрут и перенаправляют пакеты конечному получателю.

Маршрутизация на хостах IPv4

При выборе направления передачи пакета хосты используют упрощенную логику маршрутизации. Если в проекте используются подсети (как обычно и бывает), то эта логика такова:

Ключевая тема **Двухэтапный процесс маршрутизации пакетов**

Этап 1 Если IP-адрес получателя находится в той же подсети IP, что и адрес отправителя, пакет отправляется непосредственно хосту-получателю

Этап 2 В противном случае пакет отправляется на *стандартный шлюз* (default gateway) (он же *стандартный маршрутизатор* (default router)). (Этот маршрутизатор имеет интерфейс в той же подсети, что и хост.)

Рассмотрим пример на рис. 4.10 и сосредоточимся на локальной сети Ethernet (*слева*). Когда компьютер PC1 посылает пакет IP на компьютер PC11 (150.9.1.11), он осуществляет сначала некие вычисления, связанные с подсетями. Компьютер PC1 устанавливает, что IP-адрес компьютера PC11 находится в той же подсети, что и PC1, поэтому он игнорирует свой стандартный маршрутизатор (150.9.1.1) и посылает пакет непосредственно на PC11, как показано на этапе 1 рисунка.

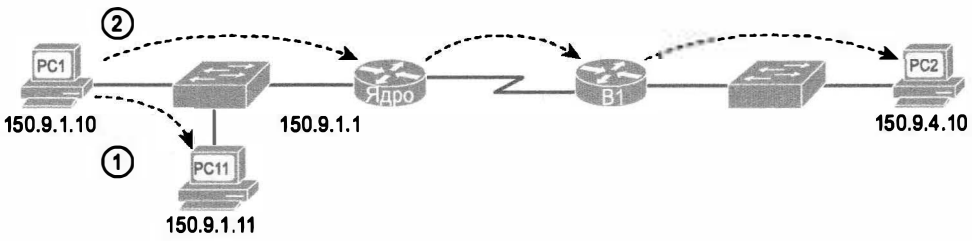


Рис. 4.10. Маршрутизация на хостах: передача хосту в той же подсети

Когда компьютер PC1 посылает пакет на компьютер PC2 (150.9.4.10), он осуществляет те же вычисления подсети и понимает, что PC2 не находится в той же подсети, что и PC1. Поэтому он перенаправляют пакет (этап 2) на свой стандартный шлюз, 150.9.1.1, который затем перенаправляет пакет компьютеру PC2.

Решение о перенаправлении и таблица маршрутизации IP

На рис. 4.1 была представлена концепция сетевого уровня маршрутизации, а на рис. 4.2 — логика инкапсуляции канала связи, связанная с маршрутизацией. Теперь рассмотрим тот же процесс на примере с тремя маршрутизаторами, перенаправляющими один пакет. Но сначала подведем итог тому, как маршрутизатор осуществляет перенаправление пакета.

Резюме логики маршрутизатора

Когда маршрутизатор получает фрейм канала связи со своим адресом, он должен обработать его содержимое. Для этого маршрутизатор применяет к фрейму канала связи следующую логику.

Четырехэтапный процесс маршрутизации пакетов

Ключевая тема

Этап 1 Для проверки ошибок фрейма используется поле контрольной суммы фрейма (FCS) канала связи. Если есть ошибки, фрейм отбрасывается

Этап 2 Если пакет не был отброшен на предыдущем этапе, отбрасывается старый канальный заголовок и концевик и остается только пакет IP

Этап 3 IP-адрес отправителя пакета IP сопоставляется с таблицей маршрутизации и определяется маршрут, который лучше всего соответствует этому адресу; маршруг идентифицирует исходящий интерфейс маршрутизатора и, возможно, IP-адрес маршрутизатора следующего перехода

Этап 4 Пакет IP инкапсулируется в новый канальный заголовок и концевик, подходящий для исходящего интерфейса, и фрейм отправляется

Согласно этим этапам, каждый маршрутизатор перенаправляет пакет следующей области, заключив его во фрейм канала связи. Каждый маршрутизатор повторяет этот процесс, пока пакет не достигнет своего конечного получателя.

Главным этапом маршрутизации является этап 3. Заголовок пакета содержит IP-адрес получателя, а таблица маршрутизации — список номеров сетей и подсетей. Чтобы найти соответствующую запись в таблице маршрутизации, маршрутизатор использует следующую логику.

Номера сети и подсети представляют группу адресов, начинающихся с того же префикса. Считайте эти номера группами адресов. Так как же выяснить, какой из групп соответствует адрес получателя этого пакета?

Конкретный пример поиска соответствия в таблице маршрутизации приведен ниже.

Пример маршрутизации

Пример маршрутизации приведен на рис. 4.11. В этом примере все маршрутизаторы используют *открытый протокол поиска первого кратчайшего маршрута* (Open Shortest Path First — OSPF) и все маршрутизаторы знают маршруты для всех подсетей. В частности, компьютер PC2 (см. рис. 4.11, *снизу*) находится в подсети 150.150.4.0, которая состоит из всех адресов, начинающихся с 150.150.4. Здесь компьютер PC1 посылает пакет IP на IP-адрес 150.150.4.10 компьютера PC2.

ВНИМАНИЕ!

В данном случае все маршрутизаторы "знают", что подсеть 150.150.4.0 означает "все адреса, которые начинаются с 150.150.4".

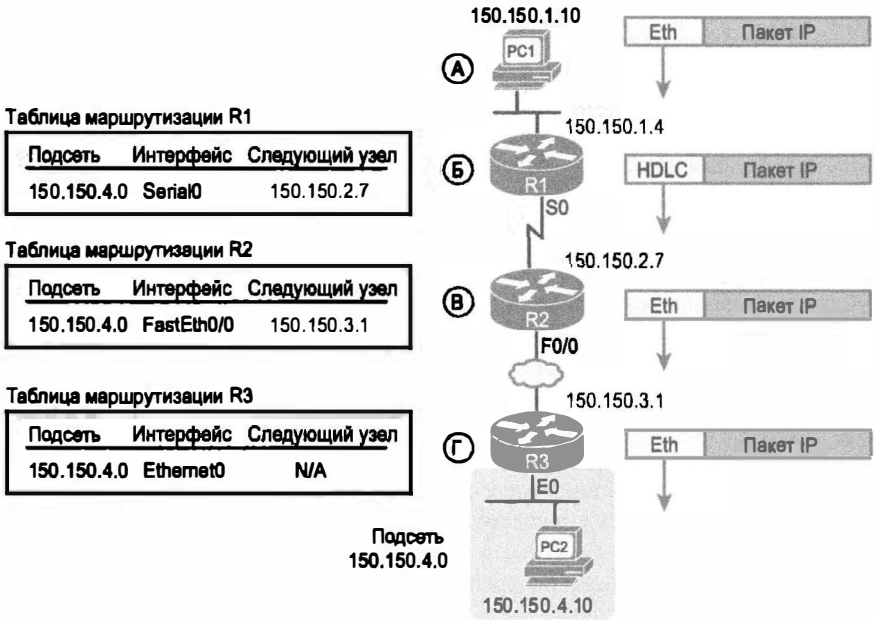


Рис. 4.11. Простой пример маршрутизации в подсетях IP

Ниже поясняется логика перенаправления на каждом этапе, показанном на рисунке. (Следует отметить, что все упоминания этапов 1–4 относятся к описанию логики маршрутизации, которое приведено выше.)

Этап А **Компьютер PC1 отправляет пакет своему стандартному маршрутизатору.** Сначала PC1 создает пакет IP, адресом получателя которого указан IP-адрес компьютера PC2 (150.150.4.10). PC1 должен отправить пакет маршрутизатору R1 (это его стандартный маршрутизатор), поскольку получатель находится в другой подсети. PC1 помещает пакет IP во фрейм Ethernet, в котором адрес Ethernet получателя соответствует адресу Ethernet маршрутизатора R1. PC1 отправляет этот фрейм по Ethernet. (Обратите внимание: концевики канала связи на рисунке не показаны.)

Этап Б **Маршрутизатор R1 обрабатывает входящий фрейм и перенаправляет пакет маршрутизатору R2.** Так как указанный во входящем фрейме Ethernet MAC-адрес получателя является MAC-адресом маршрутизатора R1, этот маршрутизатор копирует фрейм для обработки. Маршрутизатор проверяет поле FCS фрейма и убеждается, что ошибок нет (этап 1). Затем маршрутизатор отбрасывает заголовок и концевик Ethernet (этап 2). После этого маршрутизатор R1 сравнивает указанный в пакете адрес получателя (150.150.4.10) с таблицей маршрутизации и находит запись для подсети 150.150.4.0. Эта подсеть содержит адреса с 150.150.4.0 по 150.150.4.255 (этап 3). Поскольку адрес получателя находится в этой группе, маршрутизатор R1 после инкапсуляции пакета во фрейм HDLC перенаправляет пакет через исходящий интерфейс Serial0 следующему маршрутизатору R2 (150.150.2.7) (этап 4)

Этап В **Маршрутизатор R2 обрабатывает входящий фрейм и перенаправляет пакет маршрутизатору R3.** Получив фрейм HDLC, маршрутизатор R2 повторяет тот же общий процесс, что и маршрутизатор R1. Маршрутизатор R2 проверяет поле FCS и обнаруживает, что ошибок нет (этап 1). Затем он отбрасывает заголовок HDLC и концевик (этап 2). После этого R2 находит в своей таблице маршрутизации маршрут для подсети 150.150.4.0, которая включает в себя диапазон адресов 150.150.4.0–150.150.4.255, и "понимает", что адрес 150.150.4.10 соответствует этому маршруту (этап 3). Наконец, маршрутизатор R2 отправляет пакет через интерфейс Fast Ethernet 0/0 следующему маршрутизатору R3 (150.150.3.1), предварительно инкапсулировав пакет в заголовок Ethernet (этап 4)

Этап Г **Маршрутизатор R3 обрабатывает входящий фрейм и перенаправляет пакет компьютеру PC2.** Маршрутизатор R3, как и R1 и R2, проверяет поле FCS, отбрасывает старый заголовок и концевик канального уровня и находит свой маршрут для подсети 150.150.4.0. Запись для этой подсети в его (маршрутизатора R3) таблице маршрутизации показывает, что исходящим интерфейсом является интерфейс Ethernet маршрутизатора R3, но маршрутизатора следующего перехода нет, поскольку маршрутизатор R3 подключен непосредственно к подсети 150.150.4.0. Маршрутизатору R3 остается только инкапсулировать пакет в новом заголовке Ethernet и концевике, указав в качестве адреса Ethernet получателя MAC-адрес компьютера PC2, и передать фрейм

Теперь кратко рассмотрим концепции протоколов маршрутизации IP.

Протоколы маршрутизации IPv4

Процесс маршрутизации (перенаправления) полностью зависит от наличия точной и актуальной таблицы маршрутизации IP на каждом маршрутизаторе. Этот раздел представляет еще один взгляд на протоколы маршрутизации, их задачи, методы оповещения и изучения маршрутов, а также пример на основании той же объединенной сети, которая была представлена на рис. 4.10.

Сначала рассмотрим задачи протокола маршрутизации, независимо от того, как он работает.

Задачи протоколов маршрутизации IP

Ключевая тема

- Динамически определять маршруты ко всем подсетям в сети и заполнять этими маршрутами таблицу маршрутизации.
- Если доступно несколько маршрутов к какой-либо подсети, поместить в таблицу наилучший из них.
- Определять, когда маршруты в таблице оказываются неправильными, и удалять их из таблицы маршрутизации.
- Если маршрут удаляется из таблицы и доступен маршрут через соседний маршрутизатор, то добавлять такой маршрут в таблицу. (Многие считают эту цель частью предыдущей.)
- Как можно быстрее добавлять новые маршруты или заменять потерянные. (Время между потерей маршрута и нахождением работоспособной замены ему называется *временем конвергенции* (convergence time).)
- Предотвращать маршрутные петли.

Все протоколы маршрутизации используют концепцию, согласно которой маршрутизаторы могут узнавать информацию о маршрутизации друг у друга. Конечно, каждый протокол маршрутизации работает по-разному; в противном случае не было бы никакой необходимости в нескольких протоколах. Однако этапы изучения маршрутов многих протоколов маршрутизации совпадают.

Этап 1 Каждый маршрутизатор добавляет маршрут в свою таблицу маршрутизации для каждой подсети, непосредственно подключенной к этому маршрутизатору

Этап 2 Каждый протокол маршрутизации маршрутизатора сообщает своим соседям обо всех маршрутах в его таблице, включая непосредственно подключенные к нему маршруты, а также маршруты, о которых ему сообщили другие маршрутизаторы

Этап 3 После получения нового маршрута от соседа протокол маршрутизации IP маршрутизатора добавляет маршрут в свою таблицу маршрутизации. При этом в качестве маршрутизатора следующего перехода обычно записывается соседний маршрутизатор, от которого был получен этот маршрут

Например, на рис. 4.12 показана та же сеть, что и на рис. 4.11, но основное внимание в данном случае уделено тому, как три маршрутизатора узнают о подсети 150.150.4.0. Следует заметить, что протоколы маршрутизации выполняют больше работы, чем показано на рисунке. В данном случае важно то, как маршрутизаторы узнают о подсети 150.150.4.0.

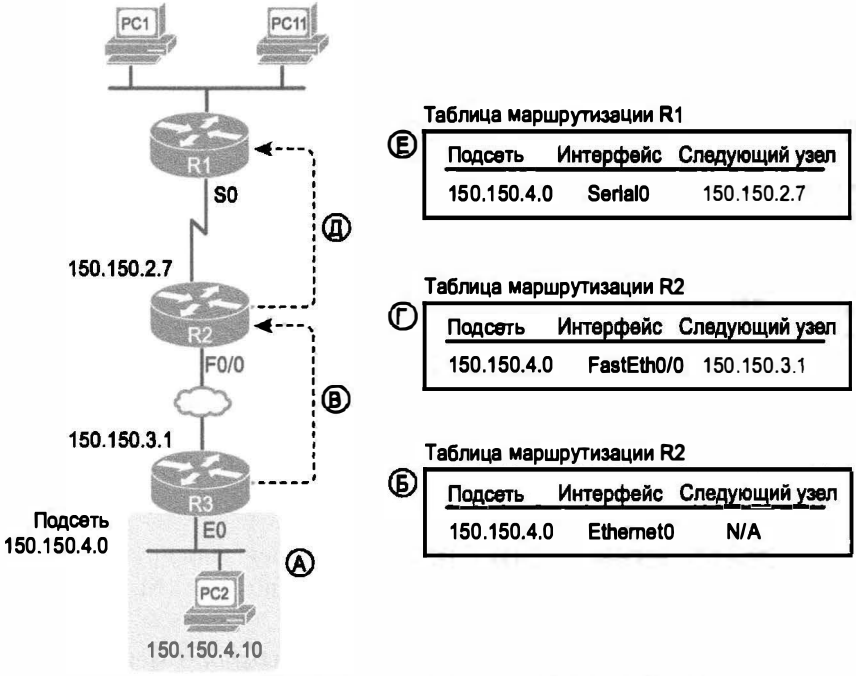


Рис. 4.12. Маршрутизатор R1 узнает о подсети 150.150.4.0

Ниже описаны этапы А, Б, В и Г, отмеченные на рисунке для того, чтобы показать, как каждый маршрутизатор изучает свой маршрут к подсети 150.150.4.0.

Этап А Подсеть 150.150.4.0, изображенная внизу рисунка, соединена с маршрутизатором R3

Этап Б Маршрутизатор R3 добавляет полученный маршрут к подсети 150.150.4.0 в свою таблицу маршрутизации IP (этап 1); это происходит без помощи протокола маршрутизации

Этап В Маршрутизатор R3 отправляет сообщение протокола маршрутизации, называемое *анонсом маршрутизации* (routing update), маршрутизатору R2, и этот маршрутизатор узнает о подсети 150.150.4.0 (этап 2)

Этап Г Маршрутизатор R3 добавляет маршрут для подсети 150.150.4.0 к своей таблице маршрутизации (этап 3)

Этап Д Маршрутизатор R2 отправляет аналогичную информацию маршрутизатору R1, сообщая ему о подсети 150.150.4.0 (этап 2)

Этап Е Маршрутизатор R1 добавляет маршрут для подсети 150.150.4.0 к своей таблице маршрутизации (этап 3). Маршрут указывает Serial0 в качестве исходящего интерфейса R1, и IP-адрес R2 (150.150.2.7) — как следующий транзитный маршрутизатор

В главе 17 протоколы маршрутизации рассматриваются более подробно. Далее в последнем важном разделе этой главы коротко описаны дополнительные функции, связанные с тем, как сетевой уровень перенаправляет пакеты от отправителя получателю через объединенную сеть.

Другие средства сетевого уровня

Сетевой уровень модели TCP/IP определяет много других функций, кроме определенных в соответствии с протоколом IPv4. Несомненно, ныне протокол IPv4 играет огромную роль в работе сети, определяя IP-адресацию и маршрутизацию. Однако документы RFC определяют также другие протоколы и стандарты, играющие важную роль в работе сетевого уровня. Например, такие протоколы маршрутизации, как OSPF, являются отдельными протоколами, определенными в отдельных документах RFC.

Данный раздел знакомит с тремя другими средствами сетевого уровня, что будет полезно при чтении остальной части книги. Эти три последние темы помогут заполнить пробелы и получить некоторое предоставление о темах, обсуждаемых далее. Вот эти темы:

- *система доменных имен* (Domain Name System — DNS);
- *протокол преобразования адресов* (Address Resolution Protocol — ARP);
- утилита `ping`.

Использование имен и системы доменных имен

Вообразите мир, где для каждого приложения нужен был бы отдельный компьютер и IP-адрес для него? Или вместо простых имен, таких как `google.com` или `facebook.com`, пришлось бы запомнить IP-адреса типа 74.125.225.5. Конечно, это не было бы удобно для пользователей и могло отвратить от использования компьютеров многих людей.

К счастью, модель TCP/IP определяет способ использования *имен хоста* (host name) для идентификации компьютеров. Пользователь либо вообще никогда не задумывается о другом компьютере, либо знает его по имени. Чтобы позволить обращаться к компьютеру по имени, протоколы должны динамически обнаруживать всю необходимую для этого информацию.

Например, когда в веб-браузере вводят имя хоста `www.google.com`, компьютер посылает пакет IP не с IP-адресом получателя `www.google.com`; он посылает его с IP-адресом, используемым веб-сервером Google. Модели TCP/IP нужен способ, позволяющий компьютеру находить IP-адрес по имени хоста, и этот способ подразумевает использование *системы доменных имен* (Domain Name System — DNS).

Предприятия используют процесс DNS для преобразования имен в соответствующие IP-адреса, как показано на рис. 4.13. В данном случае компьютер PC11 (слева) должен подключиться к серверу по имени Server1. Пользователь либо ввел имя **`Server1`**, либо запустил на компьютере PC11 некое приложение, обратившееся к этому серверу по имени. На этапе 1 компьютер PC11 посылает сообщение с запросом DNS на сервер DNS. На этапе 2 сервер DNS отсылает назад ответ DNS, содержащий IP-адрес сервера Server1. Теперь, на этапе 3, компьютер PC11 может послать пакет IP на адрес получателя 10.1.2.3, являющийся адресом сервера Server1.

Ключевая тема

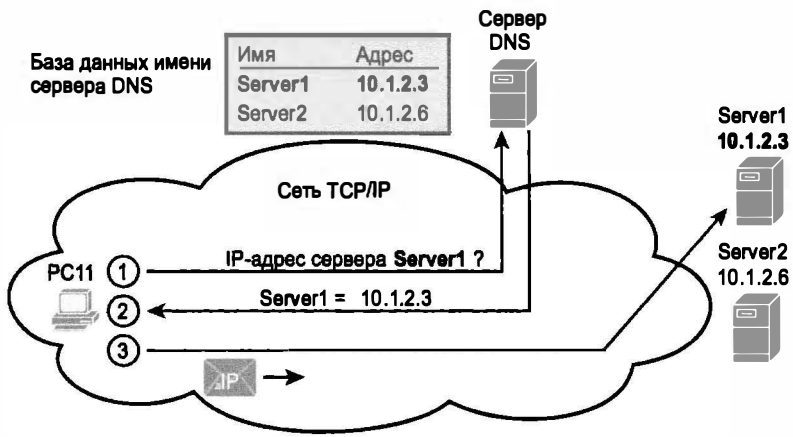


Рис. 4.13. Задача и процесс преобразования имен DNS

Обратите внимание: сеть TCP/IP на рис. 4.13 изображена в виде облака, поскольку подробности сети, включая маршрутизаторы, не имеют значения для процесса преобразования имен. Маршрутизаторы обрабатывают сообщения DNS точно так же, как любой другой пакет IP, перенаправляя их на основании IP-адреса получателя. На этапе 1 запрос DNS в качестве адреса получателя использует IP-адрес сервера DNS, используемого всеми маршрутизаторами при перенаправлении пакетов.

На самом деле система DNS делает намного больше, чем просто рассылает несколько сообщений. Она определяет протоколы, стандарты текстовых имен, используемых во всем мире, и международный набор распределенных серверов DNS. Стандартам имен DNS следуют имена доменов, которые люди ежедневно используют при просмотре веб-страниц, они выглядят как `www.example.com`. Кроме того, ни один сервер DNS сам по себе не знает всех соответствий имен IP-адресам, но

информация о них может распространяться между многими серверами DNS. Таким образом, серверы DNS всего мира взаимодействуют между собой, перенаправляя запросы друг другу, пока они не достигнут сервера, который знает ответ и предоставит необходимую информацию об IP-адресе.

Протокол преобразования адресов

Логика маршрутизации IP требует, чтобы хосты и маршрутизаторы помещали пакеты IP во фреймы канального уровня. На рис. 4.11 показано, что каждый маршрутизатор извлекает пакет IP из фрейма канала связи и помещает его в новый.

Всякий раз, когда в локальных сетях Ethernet хост или маршрутизатор должен инкапсулировать пакет IP в новый фрейм Ethernet, им известны все данные его заголовка за исключением MAC-адреса получателя. Хосту известен IP-адрес следующего устройства — это либо IP-адрес другого хоста, либо IP-адрес стандартного маршрутизатора. Маршрутизатор знает маршрут, используемый для перенаправления пакета IP, в котором значится IP-адрес следующего маршрутизатора. Однако хостам и маршрутизаторам не известны заранее MAC-адреса соседних устройств.

Чтобы любой хост или маршрутизатор в локальной сети мог динамически изучать MAC-адреса других хостов или маршрутизаторов IP в той же локальной сети, модель TCP/IP определяет *протокол преобразования адресов* (Address Resolution Protocol — ARP). Протокол ARP определяет, что включает *запрос ARP* (ARP request), т.е. сообщение, задающее простой вопрос: "Если это ваш IP-адрес, подскажите свой MAC-адрес". Протокол ARP определяет также *ответное сообщение ARP* (ARP reply), включающее первоначальный IP-адрес и соответствующий ему MAC-адрес.

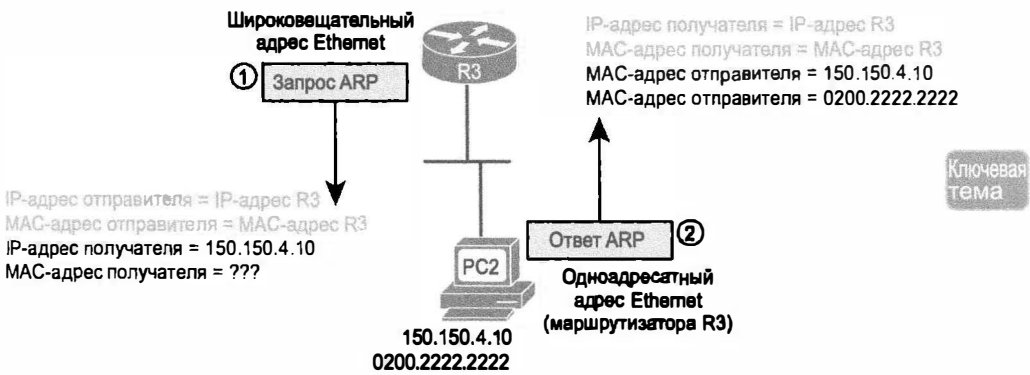


Рис. 4.14. Задача и работа процесса ARP

Обратите внимание на то, что хосты запоминают результат запроса ARP, сохраняя информацию в *кеше ARP* (ARP cache) или *таблице ARP* (ARP table). Хосты и маршрутизаторы используют протокол ARP лишь иногда, в основном для первоначального создания кеша ARP. Каждый раз, когда хост или маршрутизатор посылает пакет, инкапсулируемый во фрейм Ethernet, он сначала проверяет свой кеш ARP в поисках соответствия IP-адреса MAC-адресу. Через некоторое время хосты и маршрутизаторы удаляют записи из кеша ARP, чтобы очистить таблицу, поэтому отдельные запросы ARP впоследствии тоже возможны.

ВНИМАНИЕ!

Содержимое кеша ARP можно просмотреть в большинстве операционных систем, используя в командной строке команду **`arp -a`**.

Протокол ICMP эхо-запросов и команда `ping`

После того как сеть TCP/IP установлена, необходимо проверить базовую связь IP, не полагаясь на какие-либо приложения. Основным средством для проверки базовой сетевой связи является команда `ping`.

Утилита `ping` (Packet Internet Groper) использует *протокол управляющих сообщений Интернета* (Internet Control Message Protocol — ICMP). Она отправляет на другой IP-адрес сообщение, которое называется *эхо-запрос ICMP* (ICMP echo request). Ожидается, что компьютер с этим IP-адресом пришлет *эхо-ответ ICMP* (ICMP echo reply). Если это так, то сеть успешно проверена. Иными словами, после этого можно утверждать, что сеть может доставить пакет от одного хоста к другому и обратно. Протокол ICMP не полагается на какие-либо приложения, он лишь проверяет базовую связь IP — уровни 1—3 эталонной модели OSI (рис. 4.15).

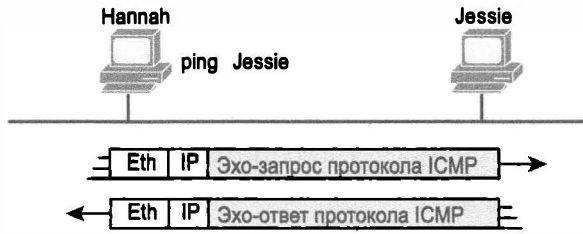


Рис. 4.15. Команда `ping` в сети

Несмотря на то что команда `ping` использует протокол ICMP, он способен на большее. Протокол ICMP определяет множество сообщений, позволяющих контролировать сеть IP. В главе 18 приведены более подробные сведения и примеры использования команды `ping` и протокола ICMP.

Обзор

Резюме

- Сетевой уровень модели TCP/IP (уровень 3) определяет правила доставки пакетов IP от первоначального устройства, создавшего пакет, на устройство его получателя.
- Протокол IP определяет логические детали передачи данных, а не физические.
- Для осуществления маршрутизации IP маршрутизаторы и компьютеры конечного пользователя (хосты сети TCP/IP) должны взаимодействовать.
- На операционной системе хоста выполняется программное обеспечение, реализующее сетевой уровень модели TCP/IP.
- Хосты используют это программное обеспечение для выбора соседнего маршрутизатора, на который следует послать пакеты IP. Этот маршрутизатор в свою очередь выбирает направление дальнейшей передачи пакета IP.
- Термин *выбор пути* иногда используется как синоним процесса маршрутизации.
- Каждый маршрутизатор хранит таблицу маршрутизации IP, содержащую группировки IP-адресов, известные также как сети и подсети IP.
- Протокол преобразования адресов (ARP) динамически изучает адрес канала связи хоста IP, подключенного к локальной сети.
- Если устройство должно общаться по протоколу TCP/IP, ему нужен IP-адрес. Когда у устройства есть IP-адрес, соответствующее программное обеспечение и оборудование могут посылать и получать пакеты IP. Любое устройство, обладающее по крайней мере одним интерфейсом с IP-адресом и способное обмениваться пакетами IP, называется хостом IP.
- IP-адреса состоят из 32-разрядного числа, обычно записанного в десятичном представлении с разделительными точками (DDN). Термин "десятичное" — свидетельство того факта, что каждый байт (8 битов) 32-разрядного IP-адреса представляется как его десятичный эквивалент. Четыре получающихся десятичных числа разделены точками (отсюда "с разделительными точками").
- Стандарт IP разделяет все пространство адресов на классы, идентифицируемые по значению первого октета.
  - Класс A включает примерно половину пространства IPv4-адресов с номерами, начинающимися с 1–126.
  - Класс B включает четверть всего пространства адресов с номерами, начинающимися с 128–191.
  - Класс C включает одну восьмую пространства адресов с номерами, начинающимися с 192–223.
  - Класс D определяет многоадресатные адреса, используемые для передачи одного пакета нескольким хостам. Их номера начинаются с 224–239.
  - Класс E определяет экспериментальные адреса с номерами, начинающимися с 240–255.

- Создание подсетей — это дальнее разделение пространства IPv4-адресов на группы размером меньше одной сети IP. Подсети IP позволяют взять одну сеть IP класса A, B или C и разделить ее на множество меньших групп последовательных IP-адресов.
- При выборе направления передачи пакета хосты используют упрощенную логику маршрутизации. Если в проекте используются подсети (как обычно и бывает), то эта логика такова:
  - если IP-адрес получателя находится в той же подсети IP, что и адрес отправителя, пакет отправляется непосредственно хосту-получателю;
  - в противном случае пакет отправляется на стандартный шлюз (он же стандартный маршрутизатор). Этот маршрутизатор имеет интерфейс в той же подсети, что и хост.
- Когда маршрутизатор получает фрейм канала связи со своим адресом, он должен обработать его содержимое. Для этого маршрутизатор применяет к фрейму канала связи следующую логику.

  Этап 1 Для проверки ошибок фрейма используется поле контрольной суммы фрейма (FCS) канала связи. Если ошибки есть, фрейм отбрасывается

  Этап 2 Если пакет не был отброшен на предыдущем этапе, отбрасывается старый канальный заголовок и концевик, а остается только пакет IP

  Этап 3 IP-адрес отправителя пакета IP сопоставляется с таблицей маршрутизации и определяется маршрут, который лучше всего соответствует этому адресу; маршрут идентифицирует исходящий интерфейс маршрутизатора и, возможно, IP-адрес маршрутизатора следующего перехода

  Этап 4 Пакет IP инкапсулируется в новый канальный заголовок и концевик, подходящий для исходящего интерфейса, и фрейм отправляется

- Модели TCP/IP нужен способ, позволяющий компьютеру находить IP-адрес по имени хоста, и этот способ подразумевает использование системы доменных имен (DNS).
- Чтобы любой хост или маршрутизатор в локальной сети мог динамически изучать MAC-адреса других хостов или маршрутизаторов IP в той же локальной сети, модель TCP/IP определяет протокол преобразования адресов (ARP).
- Утилита `ping` использует протокол управляющих сообщений Интернета (ICMP). Она отправляет на другой IP-адрес сообщение, которое называется эхо-запрос ICMP. Ожидается, что компьютер с этим IP-адресом пришлет эхо-ответ ICMP.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Что из перечисленного ниже является функциями протоколов третьего уровня модели OSI? (Выберите два ответа.)

   A) Логическая адресация (logical addressing).

   Б) Физическая адресация (physical addressing).

В) Выбор пути (path selection).

Г) Арбитраж (arbitration).

Д) Восстановление после ошибок (error recovery).

2. Предположим, компьютер PC1 должен отправить данные компьютеру PC2 и компьютеры PC1 и PC2 отделены друг от друга несколькими маршрутизаторами. Оба компьютера находятся в разных локальных сетях Ethernet. Укажите наибольший (по размеру) блок данных, который передается от PC1 к PC2. (Выберите два ответа.)

   А) Фрейм (frame).

   Б) Сегмент (segment).

   В) Пакет (packet).

   Г) L5 PDU.

   Д) L3 PDU.

   Е) L1 PDU.

   Какие из перечисленных ниже адресов являются правильными IP-адресами класса C, которые можно назначать хостам?

   А) 1.1.1.1

   Б) 200.1.1.1

   В) 128.128.128.128

   Г) 224.1.1.1

   Д) 223.223.223.255

3. Укажите диапазон значений для первого октета сетей IP класса A.

   А) от 0 до 127

   Б) от 0 до 126

   В) от 1 до 127

   Г) от 1 до 126

   Д) от 128 до 191

   Е) от 128 до 192

4. Компьютеры PC1 и PC2 находятся в двух разных сетях Ethernet, разделенных маршрутизатором IP. IP-адрес PC1 10.1.1.1 в подсети не используется. Какой из следующих адресов можно использовать для PC2? (Выберите два ответа.)

   А) 10.1.1.2

   Б) 10.2.2.2

   В) 10.200.200.1

   Г) 9.1.1.1

   Д) 225.1.1.1

   Е) 1.1.1.1

5. Представьте себе сеть с двумя маршрутизаторами, соединенными последовательным двухточечным каналом HDLC. Каждый маршрутизатор поддерживает сеть Ethernet. Компьютер PC1 подключен к сети Ethernet первого маршрутизатора (Router1), а компьютер PC2 — к сети Ethernet второго маршрутизатора (Router2). Какое утверждение справедливо при передаче данных от PC1 к PC2?

А) Маршрутизатор Router1 удаляет из фрейма, полученного от компьютера PC1, заголовок и концевик Ethernet, которые не будут использоваться.

Б) Маршрутизатор Router1 инкапсулирует фрейм Ethernet в заголовок HDLC и отправляет этот фрейм маршрутизатору Router2, который выделяет фрейм Ethernet для перенаправления к PC2.

В) Маршрутизатор Router1 удаляет из фрейма, полученного от PC1, заголовок и концевик Ethernet, восстанавливаемые маршрутизатором Router2 перед отправкой данных компьютеру PC2.

Г) Маршрутизатор Router1 удаляет заголовки Ethernet, IP и TCP и перестраивает соответствующие заголовки перед отправкой пакета маршрутизатору Router2.

6. Какие из следующих адресов обычно использует маршрутизатор, принимая решение о маршрутизации пакетов TCP/IP?

А) MAC-адрес получателя.

Б) MAC-адрес отправителя.

В) IP-адрес получателя.

Г) IP-адрес отправителя.

Е) MAC- и IP-адреса получателя.

7. Какое из приведенных ниже утверждений справедливо для подключенного к локальной сети хоста TCP/IP и его решений о маршрутизации IP? (Выберите два ответа.)

А) Хост всегда отправляет пакеты своему стандартному шлюзу.

Б) Хост всегда отправляет пакеты своему стандартному шлюзу, если IP-адрес получателя находится в сети IP другого класса.

В) Хост всегда отправляет пакеты своему стандартному шлюзу, если IP-адрес получателя находится в другой подсети.

Г) Хост всегда отправляет пакеты своему стандартному шлюзу, если IP-адрес получателя находится в той же подсети.

8. Какие из перечисленных ниже функций являются функциями протокола маршрутизации? (Выберите два ответа.)

А) Уведомление соседних маршрутизаторов об известных маршрутах.

Б) Изучение маршрутов для подсетей, непосредственно подключенных к маршрутизатору.

В) Изучение маршрутов, представленных маршрутизатору соседними маршрутизаторами, и помещение этих маршрутов в таблицу маршрутизации.

Г) Перенаправление пакетов IP на основании IP-адреса получателя пакета.

9. Компания реализует сеть TCP/IP с компьютером PC1 в локальной сети Ethernet. Какие из следующих протоколов и средств понадобятся для того, чтобы компьютер PC1 изучил информацию о некоем устройстве другого сервера?

А) ARP

Б) ping

В) DNS

Г) Ни один из ответов не правильный.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 4.6.

Таблица 4.6. Ключевые темы главы 4

Элемент	Описание	Страница
Список	Правила группировки IP-адресов в сети или подсети	142
Рис. 4.6	Распределение пространства IPv4-адресов по классам	143
Рис. 4.7	Размер частей сети и хоста у адресов сетей классов A, B и C	144
Табл. 4.5	Все возможные корректные адреса сетей	146
Рис. 4.9	Концептуальное представление подсетей	147
Список	Двухэтапный процесс маршрутизации пакетов	148
Список	Четырехэтапный процесс маршрутизации пакетов	149
Список	Задачи протоколов маршрутизации IP	151
Рис. 4.13	Задача и процесс преобразования имен DNS	154
Рис. 4.14	Задача и работа процесса ARP	155

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

стандартный шлюз и стандартный маршрутизатор (default gateway/default router), таблица маршрутизации (routing table), сеть IP (IP network), подсеть IP (IP subnet), пакет IP (IP packet), протокол маршрутизации (routing protocol), десятичное представление с разделительными точками (Dotted-Decimal Notation — DDN), IPv4-адрес (IPv4 address), одноадресатный IP-адрес (unicast IP address), создание подсетей (subnetting), имя хоста (host name), DNS, ARP, ping

Ответы на контрольные вопросы:

1 А и В. **2** В и Д. **3** Б. **4** Г. **5** Г и Е. **6** А. **7** В. **8** Б и В. **9** А и В. **10** В.

ГЛАВА 5

Основы протокола TCP/IP: передача данных и приложения

Экзамены CCENT и CCNA сосредоточены главным образом на функциях нижних уровней модели TCP/IP, определяющих то, как сети IP передают пакеты IP между хостами по локальным и глобальным сетям. Эта глава посвящена основам нескольких тем, которым на экзаменах уделяется меньше внимания: транспортный уровень и уровень приложений модели TCP/IP. Функции этих более высоких уровней играют важную роль в реальных сетях TCP/IP, поэтому, прежде чем переходить к остальной части книги, где маршрутизация и локальные сети IP рассматриваются более подробно, следует получить некоторое представление и о них.

Данная глава начинается с рассмотрения функций двух протоколов транспортного уровня: протокола управления передачей (TCP) и протокола пользовательских дейтаграмм (UDP). Второй раздел главы посвящен уровню приложений TCP/IP, включая обсуждение работы системы доменных имен (DNS).

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Наиболее распространенные приложения и их воздействие на сеть.

Передача данных между двумя хостами по сети.

Основные темы

Протоколы 4-го уровня стека TCP/IP: TCP и UDP

Транспортный уровень эталонной модели OSI (4-й уровень) определяет несколько функций, наиболее важными из которых являются восстановление данных при ошибках передачи и управление потоком. Протоколы TCP/IP транспортного уровня реализуют эти функции. Следует обратить внимание на то, что как в эталонной модели OSI, так и в модели TCP/IP этот уровень называется транспортным. Однако, как обычно, когда речь идет о модели TCP/IP, имя и номер уровня базируются на эталонной модели OSI, поэтому все протоколы TCP/IP транспортного уровня рассматриваются как протоколы 4-го уровня.

Принципиальное различие между протоколами TCP и UDP состоит в том, что по сравнению с UDP протокол TCP обеспечивает приложениям значительно более широкий диапазон служб. Например, маршрутизаторы по многим причинам могут отбрасывать пакеты, включая битовые ошибки, переполнение и ситуации, в которых правильный маршрут неизвестен. Как уже говорилось, большинство протоколов канального уровня фиксируют факт ошибки при передаче (этот процесс называется *обнаружением ошибок* (error detection)), а затем отбрасывают фреймы с ошибками. Протокол TCP обеспечивает повторную передачу (восстановление после ошибок передачи) и помогает избежать переполнения (управление потоком), в то время как протокол UDP таких функций не имеет. Вследствие этого во многих протоколах приложений предпочтительным оказывается использование протокола TCP.

Однако из-за недостатка служб в протоколе UDP не следует делать вывод, что этот протокол уступает протоколу TCP. Меньшее количество служб позволяет протоколу UDP использовать меньше байтов в заголовке, чем протоколу TCP, что приводит к уменьшению служебной нагрузки в сети. Программное обеспечение протокола UDP не вызывает замедления передачи данных в тех случаях, когда работа TCP целенаправленно замедляется. К этому следует добавить, что некоторые приложения, особенно современные технологии *передачи голоса по сети IP* (Voice Over IP — VoIP), не требуют восстановления данных после ошибок и поэтому в них используется протокол UDP. Вследствие этого протокол UDP по-прежнему играет важную роль в современных сетях TCP/IP.

В табл. 5.1 перечислены основные функции протоколов TCP и/или UDP. Следует обратить внимание на то, что протокол UDP поддерживает лишь первую функцию, а протокол TCP все перечисленные в таблице функции.

Ключевая тема

Таблица 5.1. Функции транспортного уровня модели TCP/IP

Функция	Описание
Мультиплексирование с использованием портов	Функция, позволяющая хосту-получателю по номеру порта выбрать приложение, для которого предназначены полученные данные
Восстановление после ошибок (надежность)	*Нумерация* (numbering) и подтверждение получения данных с помощью полей заголовка Sequence (Последовательный номер) и Acknowledgment (Подтверждение)

Окончание табл. 5.1

Функция	Описание
Управление потоком с использованием окон	Использование размеров окон для защиты от переполнения буфера трафиком на маршрутизаторах и хостах
Установка и прекращение соединения	Процесс инициализации номеров портов и полей Sequence (Последовательный номер) и Acknowledgment (Подтверждение)
Упорядоченная передача данных и их сегментация	Непрерывный поток байтов от процесса более высокого уровня "сегментируется" для передачи и передается процессам верхних уровней принимающего устройства с тем же порядком следования байтов

Далее описываются функции протокола TCP и выполняется их сравнение с функциями протокола UDP.

Протокол управления передачей

В каждом приложении TCP/IP обычно выбирается протокол TCP или UDP, в зависимости от требований приложения. Например, протокол TCP обеспечивает восстановление после ошибок, однако для этого требуется бо́льшая полоса пропускания и большее количество циклов обработки. Протокол UDP не осуществляет коррекцию ошибок, но требует меньшую полосу пропускания и меньшее количество циклов обработки. Независимо от того, какой из двух протоколов транспортного уровня TCP/IP выбран приложением для использования, требуется понимать базовые принципы работы каждого из этих протоколов транспортного уровня.

Протокол TCP, описанный в документе RFC 793, выполняет свои функции (см. табл. 5.1), используя механизмы конечных компьютеров. Работа протокола TCP базируется на протоколе IP для сквозной (end-to-end) доставки данных, включая вопросы маршрутизации. Иными словами, протокол TCP выполняет лишь часть функций, необходимых для доставки данных от одного приложения другому. Кроме того, выполняемая им роль направлена на обеспечение служб для приложений, которые располагаются на конечном компьютере. Независимо от того, находятся ли оба компьютера в локальной сети Ethernet или разделены огромными участками в Интернете, протокол TCP выполняет свои функции одним и тем же образом.

На рис. 5.1 показаны поля заголовка протокола TCP. Хотя нет необходимости запоминать все имена полей или их расположение, в оставшейся части раздела будут упоминаться некоторые из этих полей, поэтому в справочных целях ниже приводится весь заголовок.

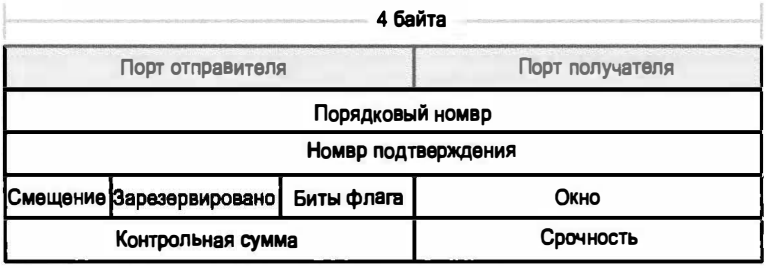


Рис. 5.1. Поля заголовка протокола TCP

Созданное протоколом TCP сообщение, начинающееся заголовком TCP и сопровождаемое прикладными данными, называется *сегментом TCP* (TCP segment). Вполне применимы также более общие термины: *PDU уровня 4* (Layer 4 PDU) или *L4PDU*.

Мультиплексирование с использованием номеров портов протокола TCP

Оба протокола — TCP и UDP — используют метод, называемый *мультиплексированием* (multiplexing). Поэтому данный раздел начинается с описания мультиплексирования в протоколах TCP и UDP. Далее будут рассмотрены функции, уникальные только для протокола TCP.

Мультиплексирование в протоколах TCP и UDP включает в себя процесс принятия решения компьютером при получении данных. Возможны ситуации, когда на компьютере работает несколько приложений, таких как веб-браузер, электронная почта или приложение VoIP (например, Skype). Мультиплексирование в протоколах TCP и UDP позволяет компьютеру, получающему данные, определять, какому из работающих приложений следует передать полученные данные.

Приведенные ниже примеры помогут понять необходимость мультиплексирования. В рассматриваемом ниже примере сеть состоит из двух персональных компьютеров (PC) — Ханны и Джесси. Компьютер Ханны использует приложение, предназначенное для рассылки рекламных сообщений, которые появляются на экране компьютера Джесси. Это приложение каждые 10 секунд посылает Джесси новое рекламное сообщение. На компьютере Ханны используется другое приложение, осуществляющее перевод денег для их пересылки компьютеру Джесси. Кроме того, на компьютере Ханны работает веб-браузер, используемый для доступа к веб-серверу, работающему на компьютере Джесси. Рекламное приложение и приложение перевода денег используются в данном примере лишь для иллюстрации. Веб-приложение работает точно так же, как это было бы в реальной жизни.

На рис. 5.2 показан пример сети, в которой на компьютере Джесси работают три приложения.

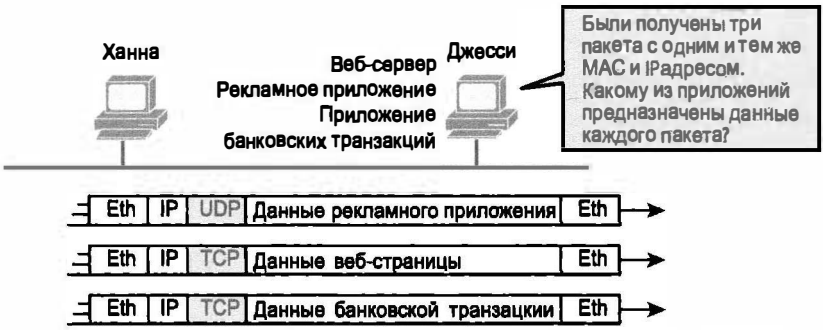


Рис. 5.2. Компьютер Ханны отправляет пакеты компьютеру Джесси с использованием трех приложений

- Рекламное приложение на основе протокола UDP.
- Приложение по переводу денег на основе протокола TCP.
- Веб-сервер на основе протокола TCP.

Компьютеру Джесси требуется знать, какому приложению следует передать данные, однако все три пакета пришли с одних и тех же адресов Ethernet и IP. Может показаться, что компьютер Джесси может решить эту проблему на основе заголовка UDP или TCP в пакете, однако, как показано на рисунке, два из трех приложений используют один и тот же протокол TCP.

Протоколы TCP и UDP решают эту проблему, используя поле номера порта в заголовке протокола TCP или протокола UDP соответственно. Каждый из сегментов TCP и UDP компьютера Ханны использует отличный от других *номер порта получателя* (destination port number), поэтому компьютер Джесси знает, какому приложению следует передать данные. На рис. 5.3 приведен пример такой ситуации.

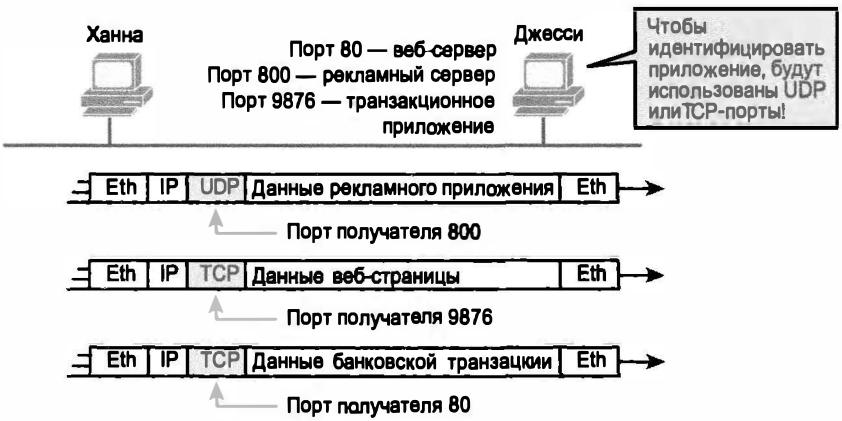


Рис. 5.3. Компьютер Ханны посылает пакеты компьютеру Джесси; при этом три приложения для мультиплексирования используют номера портов

Мультиплексирование базируется на понятии *сокета* (socket). Сокет состоит из трех частей:

- IP-адрес;
- транспортный протокол;
- номер порта.

Поэтому для веб-сервера на компьютере Джесси сокет будет иметь вид (10.1.1.2, TCP, порт 80), поскольку стандартно веб-серверы используют общеизвестный порт 80. При подключении веб-браузера на компьютере Ханны к веб-серверу также используется сокет — вероятно, имеющий вид (10.1.1.1, TCP, 1030). Почему 1030? Просто потому, что компьютеру Ханны требуется номер порта, который был бы уникальным для компьютера Ханны, а поскольку порт 1030 вполне доступен, использует его. На самом деле хосты обычно выделяют для использования динамические номера портов, начинающиеся с 1024, так как порты с номерами, меньшими 1024, зарезервированы для общеизвестных приложений.

На рис. 5.3 компьютеры Ханны и Джесси используют одновременно три приложения, следовательно, при этом открыты три сокета соединений. Поскольку сокет на каждом отдельном компьютере должен быть уникальным, соединение между двумя сокетами должно уникальным образом идентифицировать соединение между

двумя компьютерами. Эта уникальность означает, что можно использовать несколько приложений одновременно, обращаясь к приложениям, работающим на том же компьютере или на различных компьютерах. Мультиплексирование на базе сокетов обеспечивает доставку данных требуемым приложениям. На рис. 5.4 показаны три сокета соединений между компьютерами Ханны и Джесси.

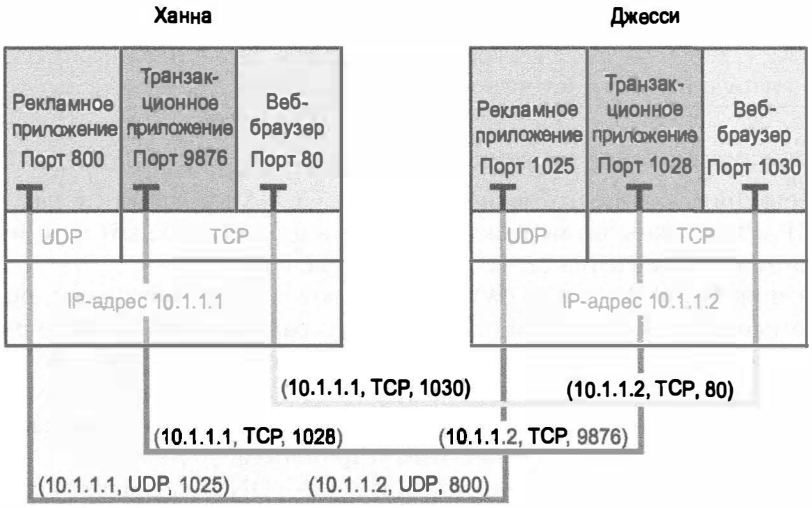


Рис. 5.4. Соединения между сокетами

Номера портов являются важнейшей частью концепции сокетов. Общеизвестные номера портов используются серверами, остальные номера портов используются клиентами. Приложения, поддерживающие службы, такие как FTP, Telnet и веб-серверы, открывают сокет, используя общеизвестные порты, и прослушивают их на предмет запросов на соединение. Поскольку от этих запросов на соединение требуется, чтобы они включали в себя номера портов как отправителя, так и получателя, номера портов, используемые серверами, должны быть общеизвестными. Поэтому стандартные службы используют специальные общеизвестные номера портов. Общеизвестные порты перечислены на странице сайта `www.iana.org/assignments/service-names-port-numbers/service-names-portnumbers.txt`.

На клиентских машинах, где инициируются запросы, для сокета может быть выделен любой неиспользуемый порт. В результате каждый клиент на том же хосте использует отличный от других номер порта, однако сервер использует для всех соединений один и тот же номер порта. Например, 100 веб-браузеров на том же компьютере могут иметь каждый свое соединение с одним и тем же веб-сервером, однако этот веб-сервер с подсоединенными к нему 100 клиентами будет использовать только один сокет и, соответственно, только один номер порта (в данном случае номер 80). Сервер может отличить пакеты любого из 100 клиентов друг от друга, просматривая порт отправителя в полученных сегментах TCP. Сервер может посылать данные требуемому клиенту (браузеру), отправляя данные на тот номер порта, который был указан в качестве порта получателя. Комбинация сокетов отправителя и получателя позволяет всем хостам, участвующим в обмене данными, отличать от-

правителей от получателей передаваемых данных. Хотя в данном примере концепция мультиплексирования была проиллюстрирована на примере 100 соединений протокола TCP, этот же метод нумерации портов относится равным образом и к сеансам протокола UDP.

ВНИМАНИЕ!

Все документы RFC можно найти на сайте по адресу `http://www.isi.edu/in-notes/rfcxxxx.txt`, где *xxxx* — номер документа RFC. Если вы не знаете номер документа RFC, выполните тематический поиск на сайте по адресу `http://www.rfc-editor.org`.

Популярные приложения протокола TCP/IP

На протяжении всей подготовки к экзамену CCNA вы встретите ряд приложений TCP/IP. Вы должны по меньшей мере знать о тех из них, которые могут быть использованы для управления сетью и контроля за ней.

Приложения World Wide Web (WWW) работают через веб-браузеры, получая таким образом содержимое, доступное на веб-серверах. Хотя WWW часто рассматривается как приложение конечного пользователя, в действительности возможно использование WWW для управления маршрутизатором или коммутатором. Для этого на маршрутизаторе или коммутаторе включается функция веб-сервера, а браузер используется для получения доступа к этим устройствам.

Система доменных имен (Domain Name System — DNS) позволяет пользователю использовать имена для ссылки на компьютеры, а служба DNS используется для нахождения соответствующих IP-адресов. Система DNS использует также модель "клиент/сервер"; при этом серверы DNS управляются сетевым персоналом, а клиентские функции DNS, как правило, являются в настоящее время частью любого устройства, использующего протокол TCP/IP. Клиент просто запрашивает у сервера DNS IP-адрес, соответствующий указанному имени.

Простой протокол управления сетью (Simple Network Management Protocol — SNMP) является протоколом уровня приложений, специально предназначенным для управления сетевыми устройствами. Например, компания Cisco поставляет на рынок разнообразные продукты управления сетью, многие из которых являются частью семейства Cisco Prime — программных продуктов управления сетью. Они могут использоваться для запроса, компиляции (сбора), хранения и отображения информации о работе сети. Для выполнения запросов к сетевым устройствам программное обеспечение Cisco Prime использует главным образом протоколы SNMP.

Для перемещения файлов на маршрутизатор, или коммутатор, или в обратном направлении программное обеспечение Cisco традиционно использовало *простейший протокол передачи файлов* (Trivial File Transfer Protocol — TFTP). TFTP определяет протокол для базовой передачи — этим и объясняется термин "простейший" (trivial). Альтернативным вариантом является использование маршрутизаторами и коммутаторами *протокола передачи файлов* (File Transfer Protocol — FTP), который имеет значительно больше функций для передачи файлов. Оба протокола успешно выполняют задачи передачи файлов на устройства Cisco и от них. Протокол FTP имеет значительно больше функций, что делает его наилучшим выбором для обычного конечного пользователя. Клиентские и серверные приложения протокола TFTP очень просты

и поэтому являются удобными инструментами в качестве встроенных частей сетевых устройств.

Одни из этих приложений используют протокол TCP, другие — протокол UDP. Например, *простой протокол передачи почты* (Simple Mail Transport Protocol — SMTP) и *почтовый протокол версии 3* (Post Office Protocol 3 — POP3), используемые для передачи электронной почты, требуют гарантированной доставки, поэтому в них используется протокол TCP. Независимо от того, какой протокол транспортного уровня используется, приложения используют общеизвестный номер порта, поэтому клиенты знают, к какому порту следует подключиться. В табл. 5.2. перечислены некоторые популярные приложения и их общеизвестные номера портов.

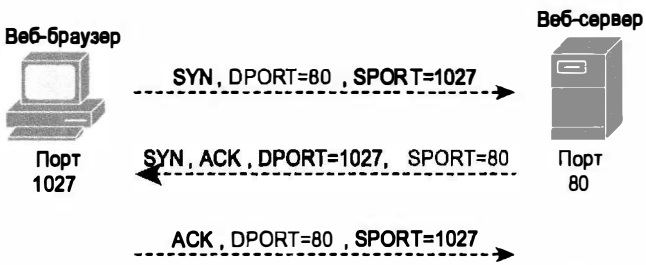
Таблица 5.2. Популярные приложения и их общеизвестные номера портов

Ключевая тема

Номер порта	Протокол	Приложение
20	TCP	Передача данных протокола FTP
21	TCP	Управление протоколом FTP
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP, TCP	DNS
67, 68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP (WWW)
110	TCP	POP3
161	UDP	SNMP
443	TCP	SSL
16 384–32 767	UDP	Передача голоса по сети IP (VoIP)

Установка и разрыв соединения

Перед тем как начнет работать какая-либо из функций протокола TCP, происходит установка соединения. Под установкой соединения понимается процесс инициализации полей последовательного номера и подтверждения, а также согласование номеров используемых портов. На рис. 5.5 приведен пример потока передаваемых данных при установке соединения.



Ключевая тема

Рис. 5.5. Установление соединения в протоколе TCP

Этот трехэтапный обмен данными при установке соединения (или *трехэтапное квитирование* (three-way handshake)) должен закончиться перед тем, как начнется передача данных. Соединение существует между двумя сокетами, хотя в заголовке TCP нет отдельного поля сокета. Предполагается, что из трех частей сокета IP-адреса могут быть получены из полей IP-адресов отправителя и получателя в заголовке протокола IP. Подразумевается протокол TCP, поскольку используется заголовок протокола TCP, который указан в поле протокола в заголовке IP. Поэтому единственной частью сокета, которая должна быть закодирована в заголовке TCP, являются номера портов.

Протокол TCP сообщает об установке соединения, используя 2 бита в полях флагов заголовка TCP. Эти биты называются флагами SYN и ACK и имеют особо важное значение. Аббревиатура SYN означает "Синхронизировать последовательные номера", что является необходимым компонентом инициализации для протокола TCP.

На рис. 5.6 показан разрыв соединения протокола TCP. Эта четырехэтапная последовательность является достаточно простой и использует дополнительный флаг, называемый *битом FIN* (FIN bit) (как, вероятно, догадался читатель, аббревиатура FIN является сокращением от "finished"). Хотелось бы сделать одно интересное замечание: перед тем как устройство, показанное на рисунке справа, отправит третий сегмент TCP в данной последовательности, оно уведомляет приложение о том, что соединение отключается. После этого оно ожидает подтверждения от приложения и лишь затем отправляет третий из показанных на рисунке сегментов. В случае, если приложению требуется некоторое время для ответа, компьютер PC справа отправляет второй из показанных на рисунке потоков данных, подтверждая то, что другой компьютер намерен разорвать соединение. В противном случае компьютер, показанный на рисунке слева, может повторно отправить первый сегмент.

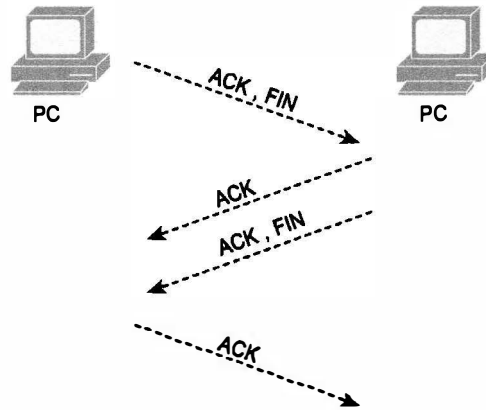


Рис. 5.6. Разрыв соединения в протоколе TCP

Протокол TCP устанавливает соединение между конечными точками и отключает его, в то время как протокол UDP этого не делает. Многие протоколы функционируют подобным образом, поэтому для обозначения каждого из этих двух подходов используются специальные термины: протоколы *с установлением соединения* (connection-oriented) и протоколы *без установления соединения* (connectionless). Более формально эти термины можно определить следующим образом.

Определения протоколов, ориентированных и не ориентированных на соединение

Ключевая тема

- *Протокол с установлением соединения.* Протокол, которому перед началом передачи данных необходим обмен сообщениями между устройствами.
- *Протокол без установления соединения.* Протокол, которому не требуется обмен сообщениями между устройствами и заранее установленной связи между конечными точками.

Протокол пользовательских дейтаграмм

Протокол UDP обеспечивает для приложений службу обмена сообщениями. В отличие от протокола TCP, этот протокол не ориентирован на соединение и не обеспечивает надежности, не использует окон и не восстанавливает порядок полученных данных. Он также не сегментирует крупные блоки данных в блоки подходящего для передачи размера. Однако протокол UDP обеспечивает выполнение некоторых функций протокола TCP, таких как передача данных и мультиплексирование с использованием номеров портов. Для этого ему требуется меньшее количество служебных байтов и меньше времени для обработки, чем протоколу TCP.

Передача данных по протоколу UDP отличается от передачи по протоколу TCP тем, что при этом не происходит упорядочение данных или их восстановление. Приложения, использующие протокол UDP, должны быть толерантны к утере данных или иметь какие-либо собственные механизмы восстановления утерянных данных. Например, в технологии VoIP используется протокол UDP, поскольку в случае утери пакета к моменту обнаружения этой потери и возможной повторной передачи накопится задержка и голос станет плохо различимым. В службе доменных имен DNS также используется протокол UDP, поскольку в случае неудачной операции по преобразованию имени пользователь может повторить попытку. Еще одним примером может служить *сетевая файловая система* (Network File System — NFS), приложение дистанционной файловой системы, которое восстанавливает данные с помощью кода уровня приложений, поэтому набор функций протокола UDP для NFS является приемлемым.

На рис. 5.7 представлен формат заголовка UDP. Обратите внимание на то, что заголовок включает поля порта отправителя и получателя, служащие той же цели, что и в протоколе TCP. Однако длина заголовка UDP составляет лишь 8 байтов, по сравнению с 20 байтами заголовка TCP, представленного на рис. 5.1. Протоколу UDP достаточно более короткого заголовка только потому, что у него меньше задач.

Рис. 5.7. Поля заголовка протокола UDP

Приложения TCP/IP

Вообще говоря, целью построения корпоративной сети, подключения небольшой домашней сети или офисной сети к Интернету является использование приложений, таких, как просмотр веб-сайтов, обмен текстовыми сообщениями, элек-

тронная почта, загрузка файлов, передача голосовых данных и видеоданных. В настоящем разделе рассматриваются некоторые вопросы проектирования сети в свете использования приложений, которые будут работать в ней. Далее будет более подробно рассмотрена работа одного конкретного приложения — веб-браузера, использующего *протокол передачи гипертекста* (Hypertext Transfer Protocol — HTTP).

Качество обслуживания в приложениях TCP/IP

Приложения должны передавать данные по объединенной сети TCP/IP. Однако они нуждаются не просто в возможности переносить данные одного приложения на одном устройстве другому приложению на другом устройстве. Используемое соединение обладает набором различных характеристик или качеств, которые сетевой мир именует *качеством обслуживания* (Quality of Service — QoS).

QoS определяет качество передачи данных между двумя приложениями и в целом по сети. Качества QoS зачастую подразделяют на четыре характеристики:

Ширина полосы пропускания (bandwidth). Количество передаваемых за секунду бит, необходимое для хорошей работы приложения; значения для приема и передачи могут быть разными или одинаковыми.

Задержка (delay). Период времени, необходимый для передачи одного пакета IP от отправителя получателю.

Дребезг (jitter). Разновидность задержки.

Потеря пакетов (loss). Процент пакетов, потерянных в сети и не поступивших получателю. При использовании протокола TCP потеря требует повторной передачи.

Сейчас в объединенных сетях TCP/IP работают приложения множества разных типов, и у каждого типа разные требования QoS. В следующих разделах рассматриваются требования QoS для трех основных категорий приложений: фоновых, интерактивных и реального времени.

Определение интерактивных и фоновых приложений

Сети TCP/IP начинались в 1970–80-е годы только с приложений данных, без всякого голоса и видео. Приложения данных посылают биты, а биты представляют данные: отображаемый пользователю текст, графические изображения, клиентскую информацию и т.д.

В зависимости от того, интерактивное ли приложение или фоновое, приложение данных предъявляет разные требования QoS. У интерактивных приложений данных на одном конце потока обычно находится пользователь (человек), а пакеты IP должны передаваться в обоих направлениях. Например, пользователь предпринимает некое действие, посылая пакет на сервер; прежде чем пользователь увидит новые данные на экране, сервер должен отослать пакет назад. Здесь задержка и дребезг оказывают большое влияние на работу пользователя.

Фоновые приложения больше сосредоточиваются на ширине полосы пропускания между двумя программными процессами. Человек в их работе, как правило, даже не участвует. Например, на устройстве может выполняться приложение создания резервной копии данных. Каждую ночь оно копирует данные на некий сервер в сети. Пользователя может вовсе не заботить то, как долго это происходит, но информационному отделу следует обеспечить достаточную ширину полосы пропуска-

ния (количество битов в секунду). Почему? Информационному отделу, возможно, придется поддерживать тысячи устройств, а резервные копии их данных следует создавать от 2:00 до 5:00 ночи каждый день.

Голосовые и видеоприложения реального времени

Большинство современных корпоративных объединенных сетей TCP/IP поддерживает также голосовые приложения. Обычно это телефоны IP — они похожи на обычные телефоны, но подключены к локальной сети TCP/IP соединением Ethernet или беспроводным соединением. Телефон передает голос как биты в пакетах IP (рис. 5.8).

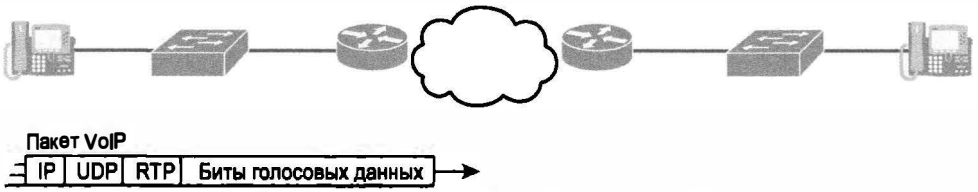


Рис. 5.8. Пакет IP, переданный при голосовой связи

ВНИМАНИЕ!

Передача голосового трафика в виде битов в пакете IP обычно упоминается как *передача голоса по сети IP* (Voice over IP — VoIP), а использование телефонов, подключенных к локальным сетям (см. рис. 5.8) — называют *телефонией IP* (IP telephony).

Хотя связь по двум телефонам IP предъявляет требования QoS, подобные интерактивным приложениям данных, но несколько выше. Например, задержка в 1 секунду вполне допустима при просмотре большинства веб-страниц, но не при качественном голосовом вызове (допустима задержка меньше 0,2 секунды). Кроме того, автономные приложения ничуть не пострадают от потери пакетов в пути, но качество передачи голоса ухудшится существенно.

Требования QoS для видео подобны таковым для голоса. Такие видеоприложения, как видеоконференция, требуют низкой задержки, низкого дребезга и низких потерь.

В табл. 5.3 сравниваются требования QoS для трех приложений: интерактивного веб-браузера, голосового и видеоприложения. Таблица со всей очевидностью демонстрирует разницу между приложениями данных и приложениями передачи голоса или видео.

Таблица 5.3. Сравнение минимальных потребностей различных типов приложений

Категория приложения	Задержка	Дребезг	Потери пакетов
Просмотр Веб-страниц (интерактивный)	Средняя	Средняя	Средняя
Вызов по телефону VoIP	Низкая	Низкая	Низкая
Видеоконференция	Низкая	Низкая	Низкая

Для поддержки требований качества обслуживания QoS различных приложений на маршрутизаторах и коммутаторах могут быть настроены различные средства обеспечения QoS. Рассмотрение этих средств выходит за рамки программы экзаменов CCNA (однако эти вопросы входят в программу нескольких сертификатов Cisco профессио-

нального уровня). Эти инструменты QoS следует использовать в современных сетях, для поддержки высокого качества служб VoIP и видео по протоколу IP.

Далее будет рассмотрен самый популярный протокол для интерактивных приложений работы с данными — протокол HTTP и "Всемирная паутина" (World Wide Web, WWW). Целью этого рассмотрения является иллюстрация работы протокола уровня приложений.

"Всемирная паутина", протоколы HTTP и SSL

"*Всемирная паутина*" (World Wide Web — WWW) включает в себя подключенные к глобальному Интернету веб-серверы всего мира и все подсоединенные к нему хосты, на которых работают веб-браузеры. Веб-сервер представляет собой службу, запущенную на некотором компьютере. Он хранит информацию (в виде веб-страниц), которая представляет интерес для многочисленных пользователей. Веб-браузер представляет собой программное обеспечение, установленное на компьютере конечного пользователя, позволяющее подключиться к веб-серверу и отобразить хранящиеся на нем веб-страницы.

ВНИМАНИЕ!

Большинство пользователей используют термин *веб-браузер*, или просто *браузер*. Веб-браузеры также называют веб-клиентами, поскольку они получают службу от веб-сервера.

Чтобы такой процесс стал возможным, должны быть доступны некоторые особые функции уровня приложений. Пользователь должен каким-либо образом указать сервер, конкретную веб-страницу и протокол, который будет использоваться для получения данных от сервера. Клиент должен узнать IP-адрес сервера, зная имя сервера; обычно для этого используется служба доменных имен DNS. Клиент должен запросить веб-страницу, состоящую из множества отдельных файлов, которые сервер должен отправить веб-браузеру. В заключение отметим, что для приложений электронной коммерции (e-commerce) передача данных, в частности конфиденциальных финансовых данных, должна сопровождаться особыми мерами безопасности, которые также обеспечиваются функциями уровня приложений. Все эти функции рассматриваются в последующих разделах данной главы.

Унифицированный локатор ресурса

Чтобы в браузере отображалась какая-либо веб-страница, он должен указать сервер, на котором она хранится, а также другую информацию, идентифицирующую конкретную веб-страницу. Как правило, на веб-сервере хранится большое количество веб-страниц. Например, если веб-браузер используется для навигации по серверу `www.cisco.com` и пользователь щелкнет мышью на соответствующей гиперссылке, то отобразится другая веб-страница. После нового щелчка отобразится еще одна новая страница. В каждом случае щелчок мышью идентифицирует IP-адрес сервера и конкретную веб-страницу; при этом большинство деталей процесса скрыто. Позиции на веб-странице, на которых можно щелкнуть мышью и вызвать другую веб-страницу, называются гиперссылками, или просто *ссылками* (link).

Пользователь браузера может указать веб-страницу, щелкнув на ее гиперссылке на отображаемой в данный момент странице, или ввести в поле адреса браузера *унифицированный локатор ресурсов* (Uniform Resource Locator — URL), зачастую на-

зываемый веб-адресом или универсальным локатором ресурсов. Оба способа — щелчок на гиперссылке или ввод URL — указывают на URL, поскольку после щелчка на ссылке веб-страницы ссылка в действительности указывает на URL.

ВНИМАНИЕ!

Большинство браузеров предоставляют какой-нибудь способ просмотреть скрытый под ссылкой URL. В одних достаточно навести курсор мыши на ссылку, в других — щелкнуть правой кнопкой мыши и выбрать в появившемся контекстном меню пункт Properties (Свойства). В появившемся окне должен отобразиться URL, к которому обратится браузер после щелчка мышью на данной гиперссылке.

Каждый URL определяет протокол, используемый для передачи данных, имя сервера и конкретную веб-страницу на этом сервере. URL можно разделить на три части:

- протокол, указываемый перед символами //;
- название хоста, которое находится между символами // и /;
- имя веб-страницы, которое находится после символа /;

Например: `http://www.certskills.com/ICND1`.

В данном случае используется *протокол передачи гипертекста* (Hypertext Transfer Protocol — HTTP), именем хоста является `www.certskills.com`, а имя веб-страницы — `ICND1`.

Поиск веб-сервера с помощью службы доменных имен DNS

Как упоминалось в главе 4, хост может использовать службу DNS для того, чтобы узнать IP-адрес, соответствующий конкретному имени хоста. Хотя URL может включать в себя IP-адрес веб-сервера вместо его имени, обычно в URL указывается имя хоста. Таким образом, перед тем как браузер сможет отправить пакет на веб-сервер, ему обычно требуется преобразовать указанное в URL имя сервера в соответствующий этому имени IP-адрес.

Чтобы обобщить вышеизложенные концепции, на рис. 5.9 показан процесс службы DNS, инициированный веб-браузером, а также приведена другая связанная с этим процессом информация. В целом можно сказать, что пользователь вводит URL (`http://www.cisco.com/go/learningnetwork`), а браузер преобразует имя `www.cisco.com` в корректный IP-адрес и начинает отправку пакетов на веб-сервер.

На рисунке показаны следующие этапы процесса.

1. Пользователь вводит URL `http://www.cisco.com/go/learningnetwork` в адресной строке браузера.
2. Клиент посылает запрос DNS на сервер DNS. Как правило, клиент узнает IP-адрес сервера DNS с помощью протокола DHCP. Отметим, что запрос DNS использует заголовок протокола UDP с портом получателя, равным общеизвестному порту 53 (список популярных общеизвестных портов см. в табл. 5.2).
3. Сервер DNS посылает ответ, приводя IP-адрес 198.133.219.25 в качестве IP-адреса для `http://www.cisco.com`. Отметим также, что в ответе содержится адрес 64.100.1.1 в качестве IP-адреса получателя, т.е. IP-адрес клиента. В нем также содержится заголовок UDP с портом отправителя 53; это вызвано тем, что данные посылаются сервером DNS.

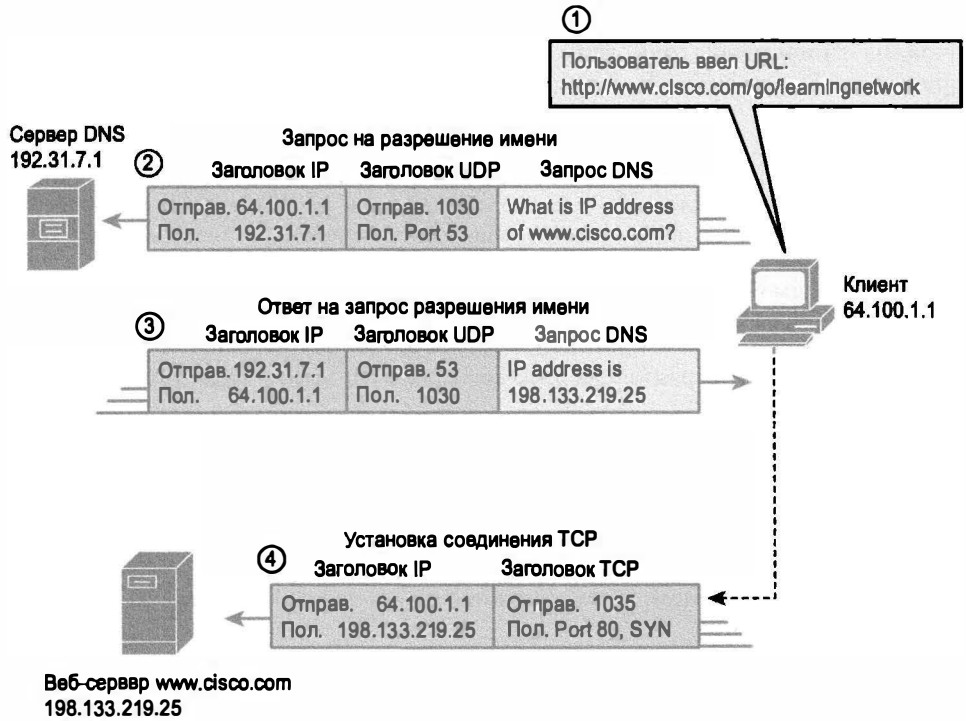


Рис. 5.9. Работа службы DNS и запрос веб-страницы

4. Клиент начинает установку нового соединения TCP с веб-сервером. Отметим, что IP-адресом получателя является только что ставший известным IP-адрес веб-сервера. Пакет включает в себя заголовок TCP, так как протокол HTTP использует протокол TCP. Отметим также, что портом TCP получателя является порт 80, т.е. общеизвестный порт HTTP. В заключение отметим, что на рисунке показан бит SYN — это напоминает о том, что процесс установки соединения TCP начинается с сегмента TCP, в котором установлен бит SYN (бинарная единица, 1).

На этой стадии процесса веб-браузер почти полностью закончил установку соединения TCP с веб-сервером. В следующем разделе будет показано, как веб-браузер получает файлы, образующие запрашиваемую веб-страницу.

Передача файлов с помощью протокола HTTP

После того как веб-клиент (браузер) создал соединение TCP с веб-сервером, клиент может запрашивать у сервера веб-страницы. Чаще всего для передачи веб-страниц используется протокол HTTP. Этот протокол уровня приложений, описанный в документе RFC 2616, определяет способ передачи файлов между двумя компьютерами. Протокол HTTP был специально создан для передачи файлов между веб-серверами и веб-клиентами.

В протоколе HTTP определено несколько команд и ответов; при этом наиболее часто используется запрос HTTP GET. Для получения файла с веб-сервера клиент посылает серверу HTTP запрос GET с указанием имени файла. Если сервер прини-

мает решение послать файл, то он отправляет ответ на запрос GET с кодом ответа 200 (означающим "OK"), а также содержимое файла.

ВНИМАНИЕ!

Для запросов HTTP имеется множество кодов ответов. Например, когда север не имеет запрашиваемого файла, он возвращает код 404, означающий "файл не найден". Большинство веб-браузеров, получая в ответ на запрос код 404, отображают пользователю не код HTTP, а сообщение вроде "страница не найдена".

Веб-страницы обычно состоят из нескольких файлов, называемых *объектами* (object). Большинство веб-страниц содержат текст, а также графические изображения, анимированную рекламу и, возможно, файлы голосовых данных или видеоданных. Каждый из этих компонентов хранится как отдельный объект (файл) на веб-сервере. Для получения всех этих файлов веб-браузер сначала получает первый файл, который может включать в себя (и обычно включает) ссылки на другие URL, поэтому браузер после этого запрашивает другие объекты. На рис. 5.10 показана общая картина этого процесса, в котором браузер получает первый файл, а затем два других файла.

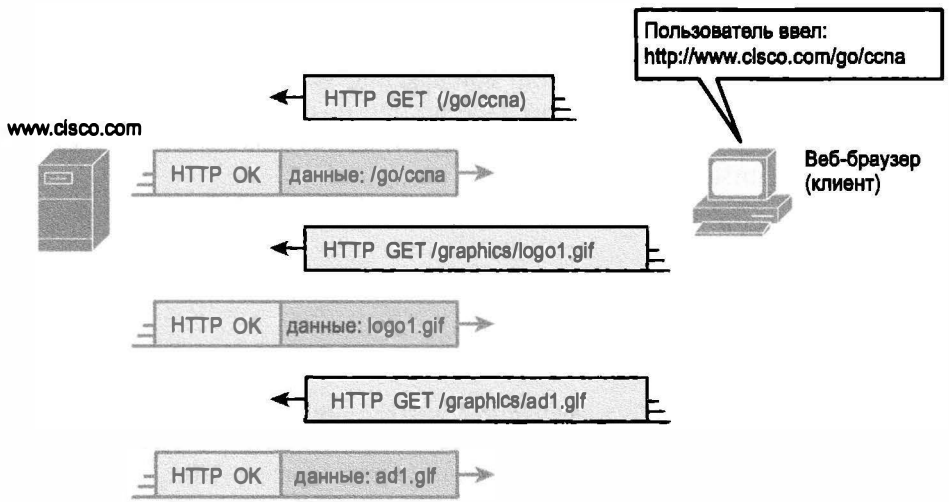


Рис. 5.10. Несколько запросов GET и ответов протокола HTTP

В данном случае после получения первого файла, обозначенного в URL как `/go/ccna`, браузер читает и интерпретирует этот файл. Кроме элементов страницы, полученный файл содержит ссылки на два других файла, поэтому браузер создает и отправляет два дополнительных запроса GET протокола HTTP. Отметим, что, хотя это и не показано на рисунке, все эти команды передаются по одному (или нескольким) соединению TCP между клиентом и сервером. Это означает, что протокол TCP обеспечивает восстановление после ошибок, гарантируя доставку данных.

Обзор

Резюме

- Транспортный уровень эталонной модели OSI (4-й уровень) определяет несколько функций, наиболее важными из которых являются восстановление данных при ошибках передачи и управление потоком. Протоколы TCP/IP транспортного уровня реализуют эти функции.
- Принципиальное различие между протоколами TCP и UDP состоит в том, что по сравнению с UDP протокол TCP обеспечивает приложениям значительно более широкий диапазон служб.
  - Например, протокол TCP обеспечивает восстановление после ошибок, но требует более широкой полосы пропускания и больше циклов обработки.
  - Протокол UDP восстановления после ошибок не обеспечивает, но требует меньшей пропускной способность и меньше циклов обработки.
- Мультиплексирование в протоколах TCP и UDP включает в себя процесс принятия решения компьютером при получении данных. Возможны ситуации, когда на компьютере работает несколько приложений, таких как веб-браузер, электронная почта или приложение VoIP (например, Skype). Мультиплексирование в протоколах TCP и UDP позволяет получающему данные компьютеру определять, какому из работающих приложений следует передать полученные данные.
- Мультиплексирование базируется на понятии сокета, состоящего из трех частей:
  - IP-адрес;
  - транспортный протокол;
  - номер порта.
- Номера портов являются важнейшей частью концепции сокетов. Общеизвестные номера портов используются серверами, остальные номера портов используются клиентами. Приложения, поддерживающие службы, такие как FTP, Telnet и веб-серверы, открывают сокет, используя общеизвестные порты, и прослушивают их на предмет запросов на соединение.
- Перед тем как начнет работать какая-либо из функций протокола TCP, происходит установка соединения. Под установкой соединения понимается процесс инициализации полей последовательного номера и подтверждения, а также согласование номеров используемых портов.
- Протокол TCP устанавливает соединение между конечными точками и отключает его, в то время как протокол UDP этого не делает. Многие протоколы функционируют подобным образом, поэтому для обозначения каждого из этих двух подходов используются специальные термины: протоколы с установлением соединения и протоколы без установления соединения.

  - *Протокол с установлением соединения.* Протокол, которому перед началом передачи данных необходим обмен сообщениями между устройствами.
  - *Протокол без установления соединения.* Протокол, которому не требуется обмен сообщениями между устройствами и заранее установленной связи между конечными точками.
- Протокол UDP обеспечивает для приложений службу обмена сообщениями. В отличие от протокола TCP, он не ориентирован на соединение и не обеспечивает надежности, не использует окон и не восстанавливает порядок полученных данных.
- В объединенных сетях TCP/IP работают приложения множества разных типов, и у каждого типа разные требования QoS.
- QoS определяет качество передачи данных между двумя приложениями и в целом по сети. Качества QoS зачастую подразделяют на четыре характеристики.
  - *Ширина полосы пропускания.* Количество бит, передаваемых за секунду, необходимое для хорошей работы приложения; значения для приема и передачи могут быть разными или одинаковыми.
  - *Задержка.* Период времени, необходимый для передачи одного пакета IP от отправителя получателю.
  - *Дребезг.* Разновидность задержки.
  - *Потеря пакетов.* Процент пакетов, потерянных в сети и не достигших получателя. При использовании протокола TCP потеря требует повторной передачи.
- В объединенных сетях TCP/IP работают приложения множества разных типов, и у каждого типа разные требования QoS. Приложения делятся на три основные категории: фоновые, интерактивные и реального времени.
  - Фоновые приложения больше сосредоточиваются на ширине полосы пропускания между двумя программными процессами. Человек в их работе, как правило, даже не участвует.
  - У интерактивных приложений данных на одном конце потока обычно находится пользователь (человек), а пакеты IP должны передаваться в обоих направлениях. Например, пользователь предпринимает некое действие, посылая пакет на сервер; прежде чем пользователь увидит новые данные на экране, сервер должен отослать пакет назад. Здесь задержка и дребезг оказывают большое влияние на работу пользователя.
  - Приложения в реальном времени (голосовые и видео) требуют низкой задержки, низкого дребезга и низких потерь.
- "*Всемирная паутина*" (WWW) включает в себя подключенные к глобальному Интернету веб-серверы всего мира и все подсоединенные к нему хосты, на которых работают веб-браузеры.
  - *Веб-сервер* представляет собой службу, запущенную на некотором компьютере. Он хранит информацию (в виде веб-страниц), которая представляет интерес для многочисленных пользователей.

  - *Веб-браузер* представляет собой программное обеспечение, установленное на компьютере конечного пользователя, позволяющее подключиться к веб-серверу и отобразить хранящиеся на нем веб-страницы.
- Пользователь браузера может указать веб-страницу, щелкнув на ее гиперссылке на отображаемой в данный момент странице, или ввести в поле адреса браузера унифицированный локатор ресурсов (URL), зачастую называемый веб-адресом или универсальным локатором ресурсов.
- После того как веб-клиент (браузер) создал соединение TCP с веб-сервером, клиент может запрашивать у сервера веб-страницы. Чаще всего для передачи веб-страниц используется протокол HTTP.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Какие из перечисленных ниже функций не являются необходимыми для протокола, который считается соответствующим 4-му уровню эталонной модели OSI?

   А) Восстановление после ошибок передачи.

   Б) Управление потоком.

   В) Сегментация данных приложений.

   Г) Преобразование из бинарной формы в формат ASCII.
2. Какие из приведенных ниже полей заголовка указывают, какому из приложений TCP/IP следует передать данные, полученные компьютером? (Выберите два ответа.)

   А) Тип сети Ethernet (Ethernet Type).

   Б) Тип протокола SNAP.

   В) Поле протокола IP.

   Г) Номер порта TCP.

   Д) Номер порта UDP.

   Е) Идентификатор (ID) приложения.
3. Какие из перечисленных ниже функций не типичны для протокола TCP? (Выберите четыре ответа.)

   А) Использование оконного механизма (windowing).

   Б) Восстановление данных после ошибок.

   В) Мультиплексирование с использованием номеров портов.

   Г) Маршрутизация.

   Д) Шифрование данных.

   Е) Упорядоченная передача данных.
4. Какая из перечисленных ниже функций поддерживается как протоколом TCP, так и протоколом UDP?

   А) Использование оконного механизма (windowing).

   Б) Восстановление после ошибок.

В) Мультиплексирование с использованием номеров портов.

Г) Маршрутизация.

Д) Шифрование данных.

Е) Упорядоченная передача данных.

5. Как называются данные, которые включают в себя заголовок протокола 4-го уровня, и данные, переданные 4-му уровню вышележащими уровнями, но не включают в себя заголовки и концевики уровней 1–3? (Выберите два ответа.)

А) L3PDUB.

Б) Блок (chunk).

В) Сегмент.

Г) Пакет.

Д) Фрейм.

Е) L4PDU.

6. Какая часть адреса URL `http://www.certskills.com/name.html` указывает имя веб-сервера?

А) `http`.

Б) `www.certskills.com`.

В) `certskills.com`.

Г) `http://www.certskills.com`.

Д) Имя файла `name.html` включает в себя имя хоста.

7. При сравнении приложения VoIP с критически важным коммерческим приложением HTTP какое из приведенных ниже утверждений точно характеризует качество обслуживания, требуемое от сети? (Выберите два ответа.)

А) VoIP требует меньшего уровня утери пакетов.

Б) Протоколу HTTP требуется меньшая полоса пропускания.

В) Протокол HTTP требует более низкого уровня флуктуации задержки.

Г) VoIP требует меньшей величины задержки.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 5.4.

Таблица 5.4. Ключевые темы главы 5

Элемент	Описание	Страница
Табл. 5.1	Функции транспортного уровня модели TCP/IP	163
Табл. 5.2	Популярные приложения и их общеизвестные номера портов	169
Рис. 5.5	Установление соединения в протоколе TCP	169
Список	Определения протоколов, ориентированных и не ориентированных на соединение	171

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

установка соединения (connection establishment), обнаружение ошибок (error detection), восстановление после ошибок (error recovery), управление потоком (flow control), прямое подтверждение (forward acknowledgment), протокол HTTP (HTTP упорядоченная передача данных (ordered data transfer), порт (port), сегмент (segment), скользящие окна (sliding windows), URL, VoIP, веб-сервер (web server)

Ответы на контрольные вопросы:

1 Г. **2** Г и Д. **3** А, Б, В, и Е. **4** В. **5** В и Е. **6** Б. **7** А и Г.

Обзор части I

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части I

Задача	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей терминов		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав только этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения к книге.

Ответы на вопросы

Ответьте на вопросы обзора этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров только этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения к книге.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если вам не все понятно, уделите время повторному изучению.

Создайте диаграмму связей терминов

Часть I этой книги содержит много терминологии. Однако по мере работы над каждой главой вы с ними освоитесь. Чем лучше вы усвоите смысл основных терминов, тем проще будет читать книгу далее.

Поскольку это первое подобное упражнение в книге, попробуйте, не заглядывая в предыдущие главы или свои записи, создать шесть диаграмм связей. В центре каждой диаграммы связей содержится номер от 1 до 6, в соответствии с рис. Ч1.1. Ваша задача такова.

- Обдумайте каждый термин, который сможете вспомнить из первой части книги.
- По каждому из этих шести номеров диаграмм связей укажите элемент, встречающийся обычно рядом с ним на рисунках. Например, к номеру 1 относится

"компьютер PC", к номеру 2 — "кабель Ethernet", подключенный к компьютеру PC1 и коммутатору, и т.д.

- Добавьте ко всем диаграммам связей все применимые к ним термины, которые сможете вспомнить. Например, термин "выделенная линия" относился бы к диаграмме связей номер 5.
- Если термин относится к нескольким диаграммам связей, добавьте его к нескольким.
- Вписав в одну из диаграмм связей все термины, которые смогли вспомнить, просмотрите разделы "Ключевые термины" в конце глав 1–5. Добавьте в диаграммы связей все забытые термины.

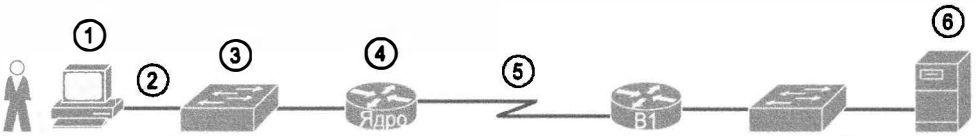


Рис. Ч1.1 Пример сети, используемой в упражнениях диаграммы связей

Задача этих диаграмм связей в том, чтобы помочь вам вспомнить термины и ассоциировать их с соответствующей частью простого проекта сети. В обзоре первой части не требуется полностью объяснить каждый термин, поскольку большинство из них подробно рассматривается только в остальной части книги.

ВНИМАНИЕ!

Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения к данной книге.

Создайте диаграммы связей из таблицы ниже на бумаге или используя любое графическое программное обеспечение. Если используется программное обеспечение, имеет смысл сохранить результат в файл для последующего анализа. Ответы приведены в приложении П (Appendix O) на веб-стайте, но ваши диаграммы связей могут выглядеть иначе.

Диаграммы связей обзора части I

Диаграмма	Описание	Где сохранен результат
1	Клиентский компьютер	
2	Канал связи Ethernet	
3	Коммутатор LAN	
4	Маршрутизатор	
5	Выделенная линия	
6	Сервер	

В части II рассматривается построение современных локальных сетей Ethernet малого и среднего размера. Глава 6 завершает обсуждение концепции LAN Ethernet, начатое во введении в локальные сети (глава 2). Главы 7 и 8 посвящены работе коммутаторов Cisco, включая возможности их администрирования и способы влияния на перенаправление ими фреймов Ethernet. Глава 9 посвящена виртуальным локальным сетям (VLAN), широко распространенному инструменту, используемому набором коммутаторов для создания множества различных локальных сетей. И наконец, глава 10 посвящена решению проблем в локальных сетях Ethernet.

Часть II. Коммутация в локальных сетях

Глава 6. “Построение локальных сетей на базе коммутаторов”

Глава 7. “Работа с коммутаторами Cisco”

Глава 8. “Настройка коммутаторов Ethernet”

Глава 9. “Реализация виртуальных локальных сетей”

Глава 10. “Поиск и устранение неисправностей на коммутаторах Ethernet”

Обзор части II

Глава 6

Построение локальных сетей на базе коммутаторов

В то время как технология Ethernet определяет происходящее в каждом канале связи Ethernet, куда более существенные события происходят на подключенных к ним устройствах: сетевых платах и коммутаторах LAN. Эта глава продолжает тему основ локальных сетей Ethernet, начатую в главе 2, более подробным освещением аспектов современных сетей Ethernet и основного устройства, используемого при создании локальных сетей, — коммутаторов LAN.

В настоящей главе обсуждение локальных сетей Ethernet и коммутации разделено на два раздела. В первом, главном, разделе рассматривается логика, используемая коммутаторами LAN при перенаправлении фреймов Ethernet, а также соответствующая терминология. Во втором разделе речь пойдет о проблемах проектирования и реализации новой локальной сети Ethernet в здании или на территории. Второй раздел посвящен таким проблемам проектирования, как использование коммутаторов в различных целях, выбор типа канала связи Ethernet и использование автопереговоров Ethernet.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы.

Выбор компонентов сети, удовлетворяющих заданной спецификации.

Передача данных между двумя хостами по сети.

Выбор подходящей среды, кабелей, портов и разъемов для подключения сетевых устройств Cisco к другим сетевым устройствам и хостам в сети LAN.

Технологии коммутации сетей LAN

Технологии и методы управления доступом к передающей среде для сети Ethernet.

Базовые концепции коммутации и работа коммутаторов Cisco.

- Домены коллизий.
- Широковещательные домены.
- Типы коммутации.
- Таблица CAM.

Основные темы

Концепции коммутации в локальных сетях

Коммутаторы получают фреймы Ethernet на одном порту, а затем перенаправляют (коммутируют) их на один или несколько других портов. Этот раздел посвящен тому, как коммутаторы принимают решение о коммутации. Кроме того, здесь рассматривается несколько сопутствующих концепций, знание которых необходимо для более полного понимания перенаправления коммутаторами фреймов Ethernet.

Чтобы современная терминология стала понятней, в этом разделе обсуждаются сначала прежние локальные сети Ethernet, использовавшие концентраторы. Основная часть данного раздела посвящена основам логики перенаправления, а завершается раздел обсуждением возможностей коммутаторов от Cisco Systems по внутренней обработке фреймов Ethernet.

Развитие сетевых устройств: концентраторы, мосты и коммутаторы

Сначала вспомним первый ориентированный на протокол UTP стандарт Ethernet — 10BASE-T, введенный в 1990 году. Стандарт 10BASE-T использовал централизованную модель кабельной проводки, подобную нынешним локальным сетям Ethernet, где каждое устройство подключено к сети LAN при помощи кабеля UTP. Однако вместо коммутатора LAN ранние сети 10BASE-T использовали концентраторы, поскольку коммутаторов LAN еще не было. Типичная топология сети 10BASE-T с концентратором приведена на рис. 6.1.

Рис. 6.1. Пример топологии сети 10BASE-T (с концентратором)

Хотя технология 10BASE-T была заметным шагом вперед в развитии сетевых технологий, у нее все же было несколько существенных недостатков, связанных с использованием концентраторов:

- получая электрический сигнал в одном порту (см. этап 1 на рис. 6.1), концентратор повторяет его на всех других портах (этап 2);
- когда два устройства или более одновременно посылают сигнал, происходит коллизия, нарушающая оба сигнала;
- в результате устройства должны соблюдать очередь, используя логику множественного доступа с контролем несущей и обнаружением конфликтов (CSMA/CD). Так устройства совместно используют ширину полосы пропускания (на 10 Мбит/с);
- широковещательные фреймы, отправляемые одним устройством, будут получены и обработаны всеми устройствами в локальной сети;
- одноадресатные фреймы получают и все остальные устройства в локальной сети.

Со временем производительности сетей Ethernet на основании концентраторов оказалось недостаточно. Разрабатывались новые приложения, требовавшие все большей ширины полосы пропускания LAN. В каждой сети Ethernet становилось все больше устройств. Однако устройства в той же сети Ethernet не могли (все вместе) передавать трафик больше, чем 10 Мбит/с, поскольку ширину полосы пропускания в 10 Мбит/с они использовали совместно. Кроме того, увеличение объемов трафика приводило к увеличению количества коллизий, что требовало более частых повторных передач и приводило к напрасной трате и так недостаточной пропускной способности LAN.

Прозрачные мосты (transparent bridge) Ethernet, или просто *мосты* (bridge), помогли решить проблему производительности сети 10BASE-T. После их добавления в локальной сети 10BASE-T произошли следующие усовершенствования:

- мосты разделяли устройства на группы, известные как *домен коллизий* (Collision Domain — CD);
- мосты сокращали количество коллизий в сети, поскольку фреймы в одном домене коллизий не вступали в конфликт с фреймами в другом;
- мосты расширяли общую полосу пропускания, предоставляя каждому домену коллизий собственную полосу пропускания (в каждом домене коллизий был только один отправитель, но в каждом свой).

На рис. 6.2 представлен переход от сети 10BASE-T без моста (см. рис. 6.1) к сети с мостом. Мост в данном случае разделяет сеть на два отдельных домена коллизий.

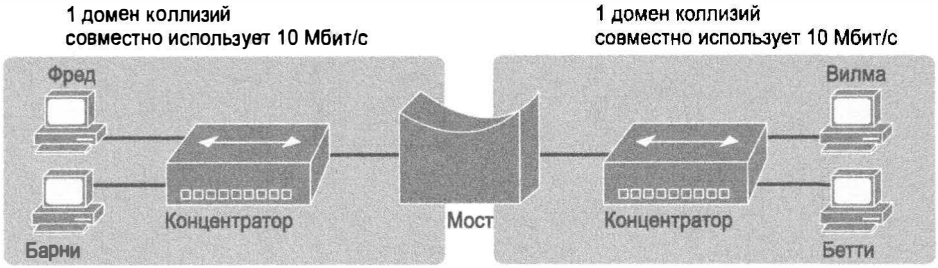


Рис. 6.2. Мост создает два домена коллизий и две разделяемые сети Ethernet

Мост (предшественник современных коммутаторов локальной сети Ethernet) использует логику, согласно которой фреймы в одном домене коллизий не вступают в конфликт с фреймами в другом. Мост передает фреймы между своими двумя интерфейсами, но в отличие от концентратора он буферизирует фреймы или ставит их в очередь, пока исходящий интерфейс не сможет их переслать. Например, Фред и Бетти одновременно послали фреймы Барни. Мост поставит фрейм от Бетти в очередь, пока домен коллизий слева не закончит передачу и не освободится.

Добавление моста на рис. 6.2 создает две отдельные сети 10BASE-T (одна слева, одна справа). Сеть 10BASE-T слева совместно использует свои 10 Мбит/с, а сеть справа свои. В результате общая ширина полосы пропускания сети в этом примере составит 20 Мбит/с. Она удвоится по сравнению с сетью 10BASE-T на рис. 6.1, поскольку устройства на каждой стороне моста могут одновременно использовать свои 10 Мбит/с.

Коммутаторы LAN выполняют те же действия, что и мосты, но для более высоких скоростей и со многими дополнительными возможностями. Как и мосты, коммутаторы сегментируют сеть LAN на отдельные домены коллизий, у каждого из которых собственная пропускная способность. Если в сети нет концентратора, каждый канал связи считается отдельным доменом коллизий, даже если никакие коллизии в нем фактически невозможны.

На рис. 6.3 приведен пример простой локальной сети с коммутатором и четырьмя компьютерами. Коммутатор создает четыре домена коллизий, способных передавать по 100 Мбит/с по каждому из четырех каналов связи. Без концентраторов каждый канал связи может работать в дуплексном режиме, удваивая пропускную способность каждого канала связи.

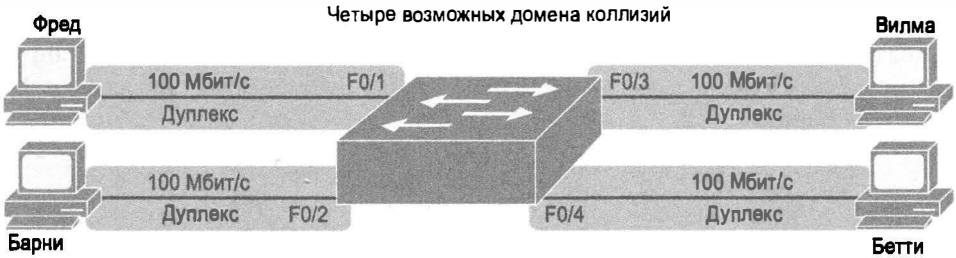


Рис. 6.3. Мост создает два домена коллизий и два разделяемых сегмента Ethernet

Логика коммутации

Вполне очевидно, что основная задача коммутатора в локальной сети состоит в пересылке фреймов Ethernet. Для этого устройство использует определенные алгоритмы, основанные на анализе MAC-адресов отправителя и получателя в заголовках Ethernet фреймов.

В этой книге рассматривается перенаправление коммутаторами одноадресатных и широковещательных фреймов, а многоадресатные фреймы Ethernet игнорируются. У одноадресатных фреймов один адрес получателя; эти адреса представляют одно устройство. У широковещательного фрейма MAC-адрес получателя имеет значение FFFF.FFFF.FFFF; такой фрейм должен быть доставлен на все устройства локальной сети.

Коммутаторы LAN получают фреймы Ethernet, а затем принимают решение о коммутации: перенаправить фрейм на некий другой порт (порты) или проигнорировать его. Для этого они выполняют три действия.

Ключевая тема

Действия, выполняемые коммутаторами

1. На основании MAC-адреса устройства получателя принимается решение переслать фрейм или отфильтровать его (не пересылать).
2. На основании MAC-адресов устройств отправителей фреймов изучаются MAC-адреса и строится таблица коммутации.
3. За счет использования *протокола распределенного связующего дерева* (Spanning Tree Protocol — STP) поддерживается топология второго уровня без петель с другими коммутаторами.

Первое из указанных выше действий — это основная задача любого коммутатора, два остальных являются второстепенными, но необходимыми. В последующих разделах подробно описаны указанные выше главные функции коммутаторов.

ВНИМАНИЕ!

При обсуждении коммутаторов LAN в этой книге термины *порт коммутатора* (switch port) и *интерфейс коммутатора* (switch interface) являются синонимами.

Фильтрация и передача фрейма

Чтобы принять решение о том, следует ли пересылать фрейм, коммутатор использует динамически создаваемую таблицу коммутации, в которой содержатся MAC-адреса и идентификаторы выходных интерфейсов. Чтобы принять решение о том, следует ли передать фрейм дальше или проигнорировать его, коммутатор сравнивает MAC-адрес получателя фрейма с записью в такой таблице. Например, обратимся к сети на рис. 6.4 и представим, что компьютер Фреда пересылает фрейм компьютеру Барни.

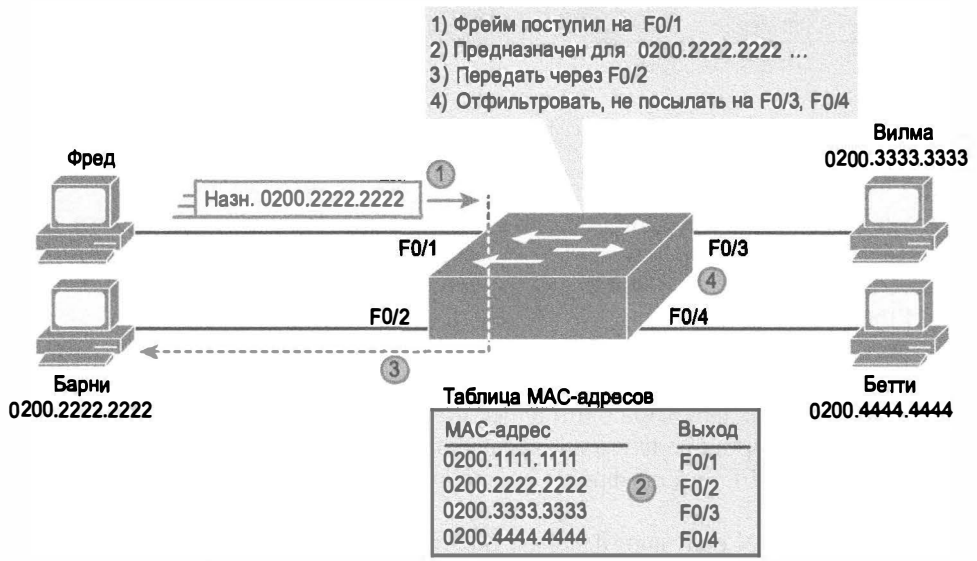


Рис. 6.4. Пример коммутации и фильтрации фреймов

На этом рисунке Фред посылает фрейм с адресом получателя 0200.2222.2222 (MAC-адрес Барни). Коммутатор сравнивает MAC-адрес получателя (0200.2222.2222) с таблицей MAC-адресов и находит соответствующую запись. Найденная запись таблицы указывает коммутатору перенаправить фрейм только на порт F0/2.

ВНИМАНИЕ!

Таблицу MAC-адресов коммутатора называют также *таблицей коммутации* (switching table), *мостовой таблицей* (bridging table) и даже *таблицей CAM* (Content Addressable Memory — *память, адресуемая по содержимому*). Последний термин обычно используется для указания типа памяти, используемой для хранения коммутационной информации.

Таблица MAC-адресов коммутатора хранит положение каждого MAC-адреса относительно данного коммутатора. В локальных сетях с несколькими коммутаторами каждый из них независимо принимает решение о перенаправлении на основании собственной таблицы MAC-адресов. Все вместе они перенаправляют фрейм так, чтобы он в конечном счете достиг места назначения.

Пример на рис. 6.5 демонстрирует те же четыре компьютера, что и рис. 6.4, но коммутаторов LAN теперь два. В данном случае Фред посылает фрейм Вилме на MAC-адрес получателя 0200.3333.3333. Согласно своей таблице MAC-адресов, коммутатор SW1 посылает фрейм на свой порт G0/1. На этапе 2 коммутатор SW2 передает фрейм на свой интерфейс F0/3, уже согласно своей таблице MAC-адресов.

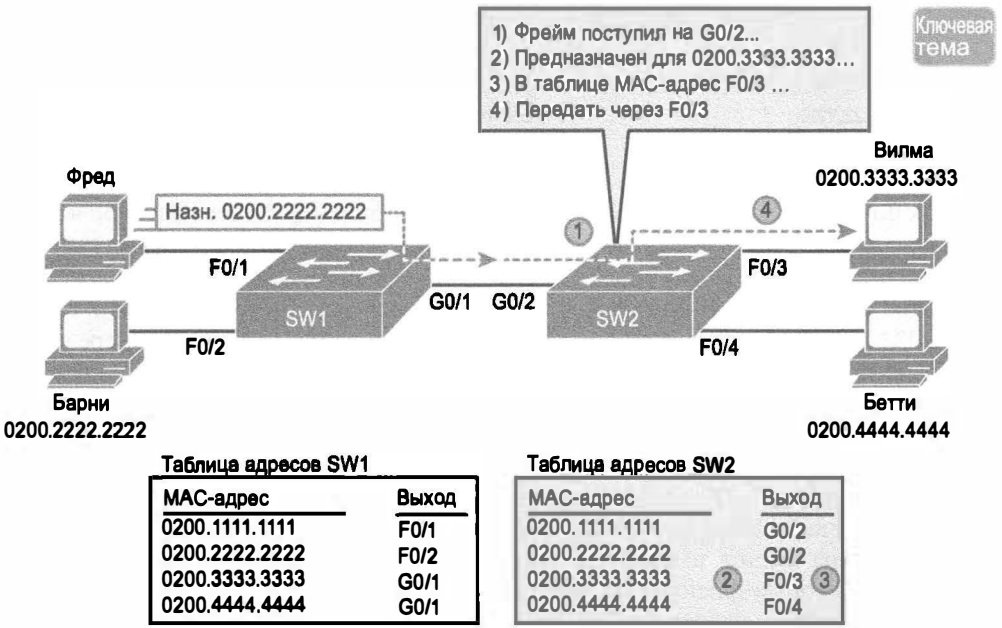


Рис. 6.5. Решение о перенаправлении при двух коммутаторах

Как коммутатор находит MAC-адреса

Другой важной функцией коммутатора является механизм обнаружения MAC-адресов и построение таблицы коммутации для них. Если таблица коммутации устройства правильная и точная, коммутатор будет принимать правильные и точные решения об отправке или фильтрации фреймов.

Коммутаторы строят таблицу адресов, просматривая входящие фреймы и выписывая из них MAC-адреса получателей. Если на вход какого-либо порта устройства получен фрейм и MAC-адрес в поле отправителя фрейма отсутствует в таблице коммутации, коммутатор создает соответствующую ему запись в таблице. В запись помещается адрес и идентификатор интерфейса, через который был получен фрейм.

На рис. 6.6 показана сеть, подобная представленной на рис. 6.4, но в этом примере коммутатор еще не построил себе таблицу коммутации, она пустая. В рассматриваемом на рис. 6.6 случае показаны два начальных фрейма, передаваемых между двумя

устройствами, первый фрейм от Фреда компьютеру Барни и ответ компьютера Барни компьютеру Фреда.

Ключевая тема

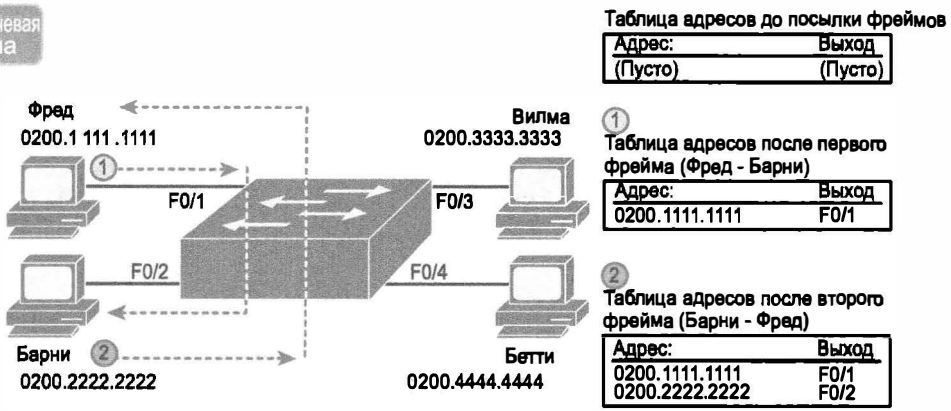


Рис. 6.6. Самообучение коммутатора: пустая таблица и добавление двух записей

Как показано на рис. 6.6, после того как компьютер Фреда переслал первый фрейм (обозначен как "фрейм 1") компьютеру Барни, коммутатор добавляет запись для MAC-адреса 0200.1111.1111 в таблицу коммутации и связывает ее с интерфейсом F0/1. Когда на втором этапе компьютер Барни пересылает ответный фрейм, его адрес, 0200.2222.2222, вместе с идентификатором интерфейса F0/2, т.е. порта, через который получен фрейм, добавляется в таблицу коммутации устройства. Коммутатор всегда использует для построения таблицы адресов MAC-адрес отправителя фрейма.

В каждой записи таблицы MAC-адресов коммутаторы хранят значение *таймера бездействия* (inactivity timer). Для новых записей коммутатор устанавливает таймер на нулевое значение. Каждый раз, когда коммутатор получает следующий фрейм с тем же MAC-адресом отправителя, он сбрасывает таймер в нуль. Со временем значение таймера растет и коммутатор получает возможность выявить записи устройств, от которых фреймы не поступали очень давно. Окончательно устаревшие записи коммутатор удаляет из таблицы. Когда у коммутатора исчерпывается место для записей в таблице MAC-адресов, он начинает удалять записи таблицы с самыми большими таймерами бездействия.

Лавинная рассылка фреймов

Теперь вернемся к процессу перенаправления на рис. 6.6. Что будет, если на коммутатор поступит первым фрейм от Фреда, когда в таблице MAC-адресов никаких записей нет? В этом случае, а также когда в таблице нет соответствующей записи, коммутатор передает фрейм на все интерфейсы (кроме входящего), используя процесс *лавинной рассылки* (flooding).

Коммутаторы рассылают *одноадресатные фреймы с неизвестным получателем* (unknown unicast frame), представляющие собой фреймы, MAC-адресов получателя которых еще нет в таблице адресов. Лавинная рассылка означает, что коммутатор перенаправляет копии фрейма на все порты, кроме того, на который он был получен. Если неизвестное устройство получит фрейм и пришлет ответ, то его MAC-адрес позволит коммутатору создать правильную запись в таблице MAC-адресов для этого устройства.

Коммутаторы также перенаправляют широковещательные фреймы, поскольку это позволяет доставить их копии на все устройства в локальной сети.

Например (см. рис. 6.6), первый посланный на MAC-адрес Барни фрейм следует только к компьютеру Барни. Но в действительности коммутатор рассылает этот фрейм на порты F0/2, F0/3 и F0/4, хотя адрес 0200.2222.2222 (Барни) доступен только на интерфейсе F0/2. Коммутатор не отправляет фрейм назад, на интерфейс F0/1, поскольку коммутатор никогда не перенаправляет фрейм на тот же интерфейс, с которого он прибыл.

Защита от петлевых маршрутов с помощью протокола STP

Третья главная функция коммутаторов локальных сетей заключается в предотвращении петлевых маршрутов с помощью *протокола распределенного связующего дерева* (Spanning Tree Protocol — STP). Без протокола STP фреймы могут бесконечно долго курсировать по петлевому маршруту, если в сети Ethernet есть резервные каналы. Чтобы избежать зацикливания фреймов, протокол STP блокирует некоторые порты, и они не могут пересылать данные. При этом в сети между сегментами (т.е. доменами коллизий) остается только один активный маршрут для передачи данных.

Результат работы протокола STP очевиден и прост: фреймы не следуют по кругу бесконечно, локальная сеть становится пригодна для использования. Однако у протокола STP есть и недостатки, включая необходимость дополнительных действий по балансировке трафика, следующего по альтернативным резервным каналам связи.

Простой пример сделает более очевидной потребность в протоколе STP. Помните: коммутатор лавинно рассылает фреймы, посланные на неизвестные одноадресатные MAC-адреса и широковещательные адреса. На рис. 6.7 представлен одноадресатный фрейм с неизвестным получателем, посланный Ларри Бобу и попавший в бесконечный цикл, поскольку в сети есть избыточный канал, но нет протокола STP.

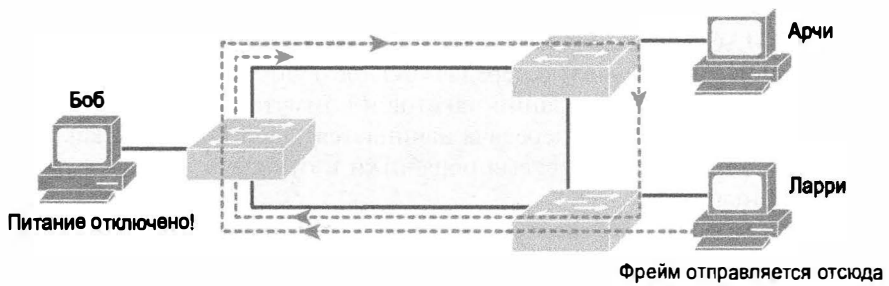


Рис. 6.7. Сеть с резервными каналами, но без протокола STP — фрейм попал в бесконечный цикл

Поскольку ни один из коммутаторов не имеет в таблицах адресов MAC-адреса Боба, каждый из них осуществляет лавинную рассылку. Физически замкнутый круг создают три коммутатора. Они продолжают перенаправлять фрейм через все порты, и его копии следуют по кругу.

Чтобы избежать петлевых маршрутов на втором уровне, на всех коммутаторах должен быть запущен протокол STP, который переводит каждый из портов каждого устройства или в *режим блокировки* (blocking state), или в *режим перенаправления* (forwarding state). Под *блокировкой* (blocking) понимают такое состояние интерфей-

са, в котором он не может передавать или принимать фреймы пользовательских данных, но может обмениваться только служебными сообщениями протокола STP. В *режиме перенаправления* интерфейс может передавать и принимать пользовательские фреймы с данными. Если протокол заблокировал правильный набор интерфейсов, в сети останется только один активный логический маршрут между каждой парой сегментов.

ВНИМАНИЕ!

Протокол STP абсолютно одинаково работает на коммутаторах и прозрачных мостах, поэтому такие термины, как *мост* (bridge), *коммутатор* (switch) и *мостовое устройство* (bridging device), используются как синонимы в описаниях протокола STP.

Более подробная информация о предотвращении циклических маршрутов при помощи протокола STP приведена в главах 1 и 2 второго тома книги.

Методы коммутации в коммутаторах Cisco

Как уже упоминалось, коммутатор принимает решение о дальнейшей передаче или фильтрации фрейма. Когда коммутатор компании Cisco принимает решение об отправке фрейма, он может использовать один из описанных ниже механизмов пересылки. На сегодняшний день подавляющее большинство устройств использует метод *коммутации с буферизацией фреймов* (store-and-forward), тем не менее все описанные ниже методы внутренней обработки потоков данных реализованы как минимум на одном из выпускаемых компанией Cisco коммутаторов.

Большинство прозрачных мостов и коммутаторов на сегодняшний день использует метод коммутации с буферизацией фреймов. В этом механизме устройство должно получить фрейм полностью, прежде чем начать отправку его первого бита через выходной интерфейс. Существуют еще два метода внутренней обработки фреймов: *коммутация без буферизации пакетов* (cut-through) и *без фрагментации* (fragment-free). MAC-адрес получателя расположен в начале заголовка Ethernet, поэтому коммутатор может начать передачу задолго до того, как он примет весь фрейм. Коммутация без буферизации пакетов и коммутация без фрагментации работают именно таким образом: передача начинается задолго до того, как будет принят весь фрейм, следовательно, время обработки и отправки (т.е. задержка (delay)) значительно уменьшается.

Метод коммутации без буферизации пакетов (cut-through) заключается в том, что устройство начинает передачу фрейма как можно раньше, т.е. как только принята часть заголовка, содержащая адрес получателя. Этот метод очень заметно уменьшает задержку в сети, но побочным его результатом будет распространение ошибок в сети. Поле *контрольной суммы фрейма* (Frame Check Sequence — FCS) находится в концевике Ethernet, следовательно, коммутатор не может определить, есть ли ошибки во фрейме, перед тем как начать передачу. Используя этот метод, следует помнить о двух основных моментах: задержка за счет обработки фреймов устройством значительно уменьшается, но за это приходится платить свою цену — фреймы с ошибками могут быть переданы дальше.

Метод коммутации без фрагментации (fragment-free processing) аналогичен методу коммутации без буферизации пакетов, но через устройство передается меньшее количество ошибок. Одна интересная особенность технологии *множественного доступа*

с контролем несущей и обнаружением коллизий (Carrier Sense Multiple Access With Collision Detection — CSMA/CD) состоит в том, что большинство коллизий происходит на первых 64 байтах фрейма. Коммутация без фрагментации похожа на коммутацию без буферизации пакетов в том смысле, что в ней также принимается только часть фрейма, 64 байта и начинается передача. Задержка за счет обработки фрейма коммутатором в таком случае будет заметно меньше, чем при буферизации, и чуть больше, чем при методе коммутации без буферизации пакетов. Количество предаваемых устройством ошибок также будет намного меньше, чем в методе коммутации с буферизацией.

На сегодняшний день большинство рабочих станций подключено к сети соединениями со скоростью 100 Мбит/с, вышестоящие каналы обычно работают на скорости 1 Гбит/с, в коммутаторах используются очень быстро работающие *специализированные микросхемы* (Application-Specific Integrated Circuits — ASIC) для аппаратной обработки потоков данных, поэтому в современных коммутаторах преимущественно используется метод коммутации с буферизацией фреймов, поскольку на таких скоростях передачи данных заметного уменьшения задержки не происходит.

Внутренние механизмы обработки фреймов в коммутаторах могут сильно отличаться у разных производителей, тем не менее все методы можно свести к трем основным или к некоторым их производным, которые перечислены в табл. 6.1.

Таблица 6.1. Методы коммутации фреймов

Ключевая тема

Метод коммутации	Описание
С буферизацией (store-and-forward processing)	Коммутатор получает фрейм полностью, до последнего бита, сохраняет, а затем начинает его передачу. Этот метод позволяет проверить целостность фрейма по контрольной сумме (FCS) до его отправки
Без буферизации (cut-through)	Коммутатор отправляет фрейм, как только будет получена нужная информация — MAC-адрес получателя. Этот метод значительно уменьшает задержку, но фреймы с неправильной контрольной суммой (FCS) не будут отброшены устройством
Без фрагментации (fragment-free processing)	Коммутатор начинает передачу, как только получит первые 64 байта фрейма. Этот метод позволяет исключить при коммутации большинство ошибочных фреймов и результатов коллизий

Резюме по коммутации в локальных сетях

В коммутаторах есть множество дополнительных функций, отсутствующих в устаревших устройствах для локальных сетей (LAN), таких как концентратор и мост. В частности, к основным достоинствам коммутаторов можно отнести перечисленные ниже.

Преимущества коммутации

Ключевая тема

- Если к порту коммутатора подключено всего одно сетевое устройство, он выполняет сегментацию сети и предоставляет выделенную полосу пропускания для устройства.

- Коммутаторы позволяют осуществлять передачу множественных одновременных потоков данных между устройствами, подключенными к разным интерфейсам.
- Если к порту коммутатора подключено всего одно сетевое устройство, работающее в дуплексном режиме, то эффективная полоса пропускания удваивается.
- Коммутаторы выполняют согласование скорости, означающее, что подключенные по технологии Ethernet устройства с разными скоростями могут взаимодействовать через коммутатор (через концентратор — не могут).

В коммутаторах используются специальные алгоритмы второго уровня, инспектирующие заголовки канального уровня для принятия решения о пересылке фрейма. В частности, коммутаторы принимают решение об отправке или фильтрации фрейма, строят таблицу MAC-адресов и используют протокол STP, чтобы разомкнуть петлевые маршруты согласно приведенной ниже последовательности.

Ключевая тема

Алгоритмы коммутации, фильтрации фреймов и построения таблиц MAC-адресов

Этап 1 Коммутаторы пересылают фреймы на основании адреса получателя в заголовке фрейма.

а) Если адрес получателя является широковещательным, много- или одноадресатным и отсутствует в таблице коммутации, то устройство лавинно рассылает фрейм.

б) Когда адрес получателя известен (т.е. присутствует в таблице MAC-адресов устройства):

1) если в таблице MAC-адресов выходной интерфейс не совпадает со входным для фрейма, коммутатор пересылает фрейм через найденный в таблице интерфейс;

2) если выходной интерфейс совпадает с входным интерфейсом, коммутатор отфильтровывает фрейм, т.е. просто игнорирует его и никуда дальше не передает

Этап 2 Коммутаторы используют следующий алгоритм для заполнения таблицы MAC-адресов.

а) Для каждого принятого фрейма считывается MAC-адрес отправителя и запоминается интерфейс, откуда он был получен.

б) Если пары "адрес–интерфейс" нет в таблице устройства, они добавляются в таблицу коммутации, таймер бездействия (inactivity timer) устанавливается в нулевое значение.

в) Если для обнаруженного MAC-адреса уже есть запись в таблице коммутации устройства, таймер сбрасывается (устанавливается в нулевое значение)

Этап 3 Коммутаторы используют протокол STP для предотвращения образования петлевых маршрутов, блокируя некоторые из интерфейсов, т.е. переключая их в такой режим, в котором они не могут принимать и передавать пользовательские фреймы

Выбор проекта локальной сети Ethernet

Первый из двух разделов этой главы посвящен подробностям работы коммутаторов LAN. Во втором разделе рассматривается множество разных тем, связанных с проектированием локальной сети.

Здесь изложено множество разных тем: концепция доменов коллизий сравнивается с концепцией широковещательных доменов. Оба подхода существенно влияют на производительность локальной сети. Эти темы должны обеспечить достаточную подготовку для последующего понимания наиболее популярного средства, упомянутого в этом разделе: виртуальной локальной сети (VLAN). Далее в этом разделе рассматриваются некоторые из проблем проектов локальных сетей Ethernet, а также способы их перевода на последние стандарты при использовании автопереговоров.

Домен коллизий, широковещательный домен и локальные сети

При создании любой локальной сети Ethernet используются коммутаторы и маршрутизаторы. Однако следует учитывать и устаревшие устройства, такие как концентраторы и мосты, — на всякий случай, если они все еще остаются в существующих сетях. В зависимости от используемых типов устройств различные части сети LAN на базе Ethernet могут вести себя по-разному с точки зрения функций и производительности. Эти различия способны повлиять на решение сетевого инженера при выборе проекта локальной сети.

Домен коллизий (collision domain) и *широковещательный домен* (broadcast domain) — термины, описывающие два важных принципа сегментации локальных компьютерных сетей с помощью оборудования разного типа. В этом разделе описано, как концентраторы, коммутаторы и маршрутизаторы влияют на границы доменов коллизий и широковещательных доменов.

Домены коллизий

Первоначально термин *домен коллизий* (collision domain) относился к концепции всех портов Ethernet, передача фреймов которых приводила к конфликту с фреймами, передаваемыми другими устройствами в домене коллизий. Рассмотрим основы концепции домена коллизий на примере рис. 6.8.

Рис. 6.8. Домены коллизий

ВНИМАНИЕ!

Структура сети, показанная на рис. 6.8, не является типичным дизайном реальной компьютерной инфраструктуры. Ее основное предназначение — проиллюстрировать домены коллизий и сравнить концентраторы, мосты, коммутаторы и маршрутизаторы.

Сначала обратите внимание на изображенные устройства. Из четырех типов сетевых устройств на рисунке только концентратор позволяет разделить устройства на домены коллизий от одной и другой стороны. Остальные сетевые устройства (маршрутизаторы, коммутаторы, мосты) разделяют локальную сеть на отдельные порты.

Современные сети физически не допускают коллизий, однако каналы связи все еще называют отдельными доменами коллизий. Рассмотрим, например, канал свя-

зи от коммутатора до компьютера 3. Физически никаких коллизий здесь произойти не может. Однако, если компьютер 3 и локальная сеть перейдут в полудуплексный режим, использующий технологию CSMA/CD, то одновременно посланные ими фреймы, в принципе, могли бы привести к коллизии. Вот почему даже сейчас домены коллизий не забыты.

Широковещательные домены

Возьмите любое устройство в любой локальной сети Ethernet и представьте, что оно передало широковещательный фрейм Ethernet. *Широковещательный домен* (broadcast domain) Ethernet — это набор устройств, получающих данный широковещательный фрейм.

Из всех сетевых устройств, представленных на рис. 6.8, только маршрутизаторы разделяют локальную сеть на несколько широковещательных доменов. Коммутаторы LAN осуществляют лавинную рассылку широковещательных фреймов Ethernet, расширяя пространство широковещательного домена. Маршрутизаторы не перенаправляют широковещательные фреймы Ethernet, они либо игнорируют их, либо обрабатывают, а затем принимают соответствующее решение. (Такие устаревшие устройства Ethernet, как мосты, действуют наподобие коммутаторов с широковещанием, а концентраторы просто повторяют сигнал, не прекращая передачи.)

На рис. 6.9 представлены широковещательные домены сети, изображенной на рис. 6.8.

Рис. 6.9. Широковещательные домены

По определению широковещательное сообщение, отправленное одним устройством в широковещательном домене, не пересылается устройствам в другом широковещательном домене. В приведенном на рис. 6.9 примере есть два широковещательных домена, т.е. маршрутизатор не будет пересылать широковещательные фреймы, отправленные компьютером, показанным на схеме слева, в сетевой сегмент, расположенный на схеме справа.

Краткие определения широковещательного домена и домена коллизий приведены ниже.

Ключевая тема **Определения широковещательного домена и домена коллизий**

- *Домен коллизий* (collision domain) — это набор плат сетевого интерфейса (NIC), в котором фрейм, посланный одной сетевой платой, может привести к коллизии с фреймом, посланным любой другой платой.
- *Широковещательный домен* (broadcast domain) — это набор плат сетевого интерфейса, получающих широковещательный фрейм, посланный любой из этих плат.

Влияние широковещательных доменов и доменов коллизий на дизайн сети

При разработке локальной сети нужно помнить основные особенности поведения сетевых интерфейсов разных устройств при подсчете их количества в каждом широковещательном домене и домене коллизий. Прежде всего следует рассмотреть поведение устройств в одном домене коллизий и помнить, что:

- устройства разделяют между собой одну доступную полосу пропускания;
- устройства могут неэффективно использовать такую полосу пропускания из-за коллизий, в частности, когда сеть сильно загружена.

С точки зрения проекта сети домены коллизий помогают сравнить старый проект, использующий концентраторы LAN, с более новым проектом, использующим те же скорости для каждого канала связи, но с коммутатором LAN вместо концентратора. Рассмотрим проект с десятью компьютерами, соединенными каналом связи 100BASE-T. При концентраторе передавать в локальной сети может только один компьютер на теоретически максимальной скорости 100 Мбит/с. Замена концентратора коммутатором даст следующее:

- пропускную способность 100 Мбит/с на каждом канале связи, а в общей сложности 1000 Мбит/с (1 Гбит/с);
- способность использовать дуплексный режим на каждом канале связи, что фактически удваивает пропускную способность до 2000 Мбит/с (2 Гбит/с).

Пример замены концентраторов коммутаторами, по общему признанию, немного устарел, но он демонстрирует одну из причин, по которой никто не использует концентраторы и производители их не производят.

Теперь рассмотрим проблемы, связанные с широковещательными сообщениями. Когда сетевой хост получает широковещательное сообщение, он обязан его обработать. Такое утверждение означает, что сетевая плата должна отправить центральному процессору (CPU) прерывание, а процессор должен выделить некоторые свои ресурсы на обработку такого фрейма. Все сетевые хосты время от времени отправляют широковещательные фреймы — они необходимы для их нормальной работы. Например, сообщения протокола ARP используют широковещательный механизм работы, как было описано в главе 4. Таким образом, широковещание будет присутствовать в любой сети, это нормальный режим работы, но широковещательные фреймы требуют затрат ресурсов всех сетевых устройств на их обработку.

Рассмотрим локальную сеть большего размера, состоящую из нескольких коммутаторов и 500 компьютеров. Все коммутаторы образуют единый широковещательный домен, поэтому если один из 500 компьютеров отправляет широковещательное сообщение, оставшиеся 499 его получают и обрабатывают. Если в такой сети много широковещательных фреймов, то они могут очень заметно снизить производительность персональных компьютеров в сети. Предположим, сеть изменилась так, что у нее стало 5 сегментов по 100 компьютеров, которые разделены маршрутизатором, следовательно, в такой сети будет 5 широковещательных доменов. При такой схеме сети широковещательное сообщение от одного компьютера будет вызывать прерывание у 99 других компьютеров, а 400 будут им "не затронуты", что приведет к меньшему снижению производительности рабочих станций.

ВНИМАНИЕ!

Построение широковещательных доменов меньшего размера в сети может также значительно улучшить безопасность сети как за счет уменьшения количества широковещательных сообщений, так и за счет расширенных функций безопасности в маршрутизаторах.

Обычно выбор между концентраторами и коммутаторами прост и заканчивается в пользу последних, решить же, когда следует использовать маршрутизаторы для разделения широковещательных доменов, намного сложнее. Подробное описание всех достоинств и недостатков разных устройств третьего уровня и возможных вариантов их реализации выходит за рамки данной книги. Тем не менее читатель должен понимать концепции широковещательных доменов, а именно, что маршрутизатор разделяет сеть на несколько широковещательных доменов, а концентратор или коммутатор — нет.

В сертификационном экзамене CCNA, скорее всего, будет много вопросов, связанных с терминологией сегментации локальных сетей и ее преимуществами, и мало простых вопросов, ориентированных просто на факты, имеющие отношение к широковещательным доменам и доменам коллизий. В табл. 6.2 перечислены основные преимущества сегментации, которые следует знать. Описанные в таблице функции можно выразить одним вопросом: "Какие из перечисленных ниже преимуществ могут быть получены при установке концентратора, коммутатора или маршрутизатора в сети Ethernet?"

Ключевая тема **Таблица 6.2. Преимущества сегментации сетей Ethernet с помощью концентраторов, коммутаторов и маршрутизаторов**

Преимущество	Концентратор	Коммутатор	Маршрутизатор
В сети общая длина кабеля может быть больше	Да	Да	Да
Сеть разделяется на множество доменов коллизий	Нет	Да	Да
Увеличивается доступная хостам полоса пропускания (bandwidth)	Нет	Да	Да
Сеть разделяется на множество широковещательных доменов	Нет	Нет	Да

Виртуальные локальные сети (VLAN)

Во многих корпоративных сетях используются *виртуальные локальные сети* (Virtual LAN — VLAN). Прежде чем перейти к описанию терминологии и принципов работы сетей VLAN, необходимо вспомнить наиболее краткое и точное определение *локальной сети* (Local Area Network — LAN). Несмотря на то что понятие локальной сети может быть знакомо читателю из других книг и источников, мы приведем наиболее точное определение с нашей точки зрения: локальной сетью (LAN) называют совокупность сетевых устройств в том же широковещательном домене.

Следовательно, первое, что нужно запомнить: без сетей VLAN коммутатор связывает все свои интерфейсы с одним широковещательным доменом. Другими словами, все подключенные к нему устройства находятся в одной локальной сети (LAN). (В комму-

таторах компании Cisco такое поведение реализовано за счет того, что все интерфейсы устройства стандартно связаны с одной сетью VLAN под номером 1.)

При использовании сетей VLAN коммутатор связывает свои интерфейсы с разными широковещательными доменами (сетями VLAN) и согласно конфигурации помещает одни интерфейсы в одну сеть VLAN, а другие — в другую. По существу, коммутатор создает несколько широковещательных доменов. Фактически все порты коммутатора формируют не единый широковещательный домен, а несколько, согласно конфигурации. Это действительно просто.

На рис. 6.10-6.11 приведены две локальные сети и показано, как на них повлияют сети VLAN. В первом примере (см. рис. 6.10), чтобы создать два раздельных широковещательных домена, приходится использовать два независимых коммутатора: по одному на каждый широковещательный домен.

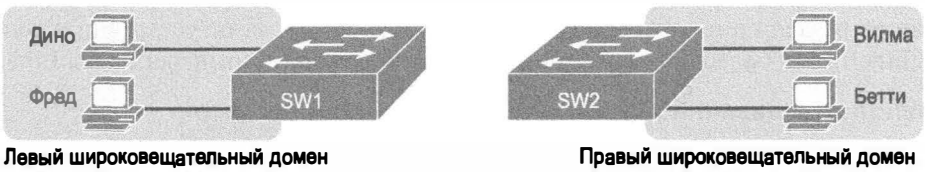
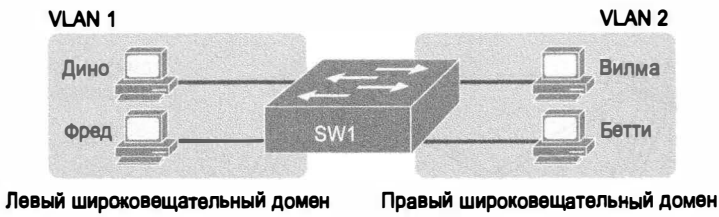


Рис. 6.10. Сеть с двумя широковещательными доменами без сетей VLAN

В качестве альтернативного решения можно предложить приведенную на рис. 6.11 схему сети с раздельными широковещательными доменами. В ней используется только один коммутатор, а раздельные широковещательные домены создаются с помощью виртуальных локальных сетей (VLAN).



Ключевая тема

Рис. 6.11. Сеть с двумя сетями VLAN и одним коммутатором

Этот раздел лишь знакомит с концепцией виртуальных сетей, более подробная информация по этой теме, включая детальное описание настройки сетей VLAN в территориальных локальных сетях, приведена в главе 9.

Выбор технологии Ethernet для территориальной локальной сети

Территориальные (или кампусные) локальные сети охватывают крупные здания или несколько рядом расположенных зданий, т.е. некоторую территорию. Так, например, компания может арендовать несколько офисных помещений в разных зданиях одного большого офисного комплекса (office park). Сетевые инженеры такой компании могут построить сеть с использованием магистральных коммутаторов в каждом здании, которые соединены между собой каналами Ethernet, и создать территориальную локальную сеть.

В процессе планирования и разработки территориальной локальной сети инженеры должны рассмотреть различные варианты технологии Ethernet и учесть максимальную длину кабеля в каждой из них. Кроме того, следует учесть различные скорости передачи данных в разных технологиях и подумать о том, что к некоторым коммутаторам будут напрямую подключены устройства пользователей, а к другим — только коммутаторы доступа к сети пользователей. Кроме всего прочего, в большинстве проектов инженеру приходится учитывать уже имеющееся в сети оборудование и оценивать, нужно ли увеличение пропускной способности и скорости существующих сегментов и следует ли покупать новое оборудование.

В этом разделе обсуждается проект территориальной локальной сети. Он начинается с рассмотрения терминологии, используемой в проектах территориальных локальных сетей. Далее следует выбор стандартов Ethernet, используемых для каналов связи территориальной локальной сети, их преимуществам и недостаткам. Завершается раздел описанием популярных средств перехода от устаревших каналов связи Ethernet на более новые (и быстрые), использующие автопереговоры Ethernet.

Терминология дизайна территориальных локальных сетей

Чтобы упорядочить все требования к территориальным локальным сетям, а также облегчить процесс обсуждения принципов дизайна сети между специалистами, в сетях, построенных на оборудовании компании Cisco, используется некоторая общая терминология для описания основных принципов и компонентов.

На рис. 6.12 представлен типичный пример проекта большой территориальной локальной сети, в которую входит примерно 1000 компьютеров, соединенных с коммутаторами на приблизительно 25 портов каждый. Объяснение терминологии приведено ниже.

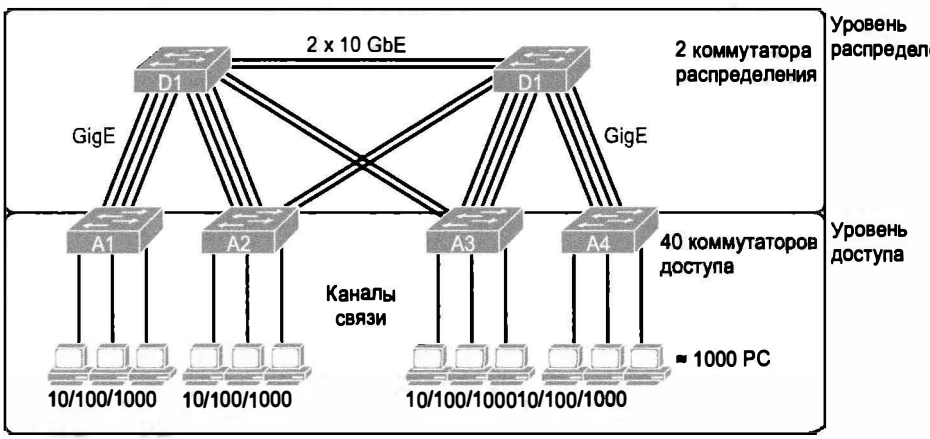


Рис. 6.12. Территориальные сети и их терминология

Компания Cisco использует три термина для описания роли каждого коммутатора в проекте территориальной сети: *уровень доступа к сети* (access), *уровень распределения* (distribution) и *ядро сети* (core). Роль зависит от того, перенаправляет ли коммутатор трафик от пользовательских устройств и остальной части локальной сети (доступ) или перенаправляет трафик между другими сетевыми коммутаторами (распределение и ядро).

Коммутаторы доступа (access switch) подключены непосредственно к конечным пользователям, предоставляя им доступ к устройствам в локальной сети. Обычно коммутаторы доступа обеспечивают обмен трафиком с подключенными к ним устройствами конечного пользователя и находятся на краю локальной сети.

В больших территориальных локальных сетях *коммутаторы распределения* (distribution switch) обеспечивают перенаправление трафика между коммутаторами доступа. В соответствии с проектом каждый из коммутаторов доступа соединен по крайней мере с одним коммутатором распределения, чтобы они могли перенаправить трафик другим частям LAN. Обратите внимание, что в большинстве проектов между коммутаторами распределения используется как минимум по два канала связи (см. рис. 6.12), для избыточности.

В проектах, где много коммутаторов доступа подключено к немногим коммутаторам распределения, снижается необходимое количество кабелей, с сохранением возможности для всех устройств передавать данные на все другие устройства локальной сети. Проект, показанный на рис. 6.12, предполагает использование двух каналов связи от каждого коммутатора доступа до каждого из двух коммутаторов распределения, обеспечивая некую избыточность. При четырех каналах связи для каждого из 40 коммутаторов доступа проект использует 160 каналов связи. Если бы этот проект не использовал коммутаторы распределения, то для соединения одним каналом связи каждой пары коммутаторов доступа потребовалось бы 780 каналов.

Самые большие территориальные локальные сети зачастую используют *коммутаторы ядра* (core switch), которые перенаправляют трафик между коммутаторами распределения. Представьте, например, небольшой городок из 12 зданий, в котором проложена локальная сеть, как на рис. 6.12. В нее можно добавить два дополнительных коммутатора ядра, играющих роль, подобную коммутаторам распределения, — они соединяют каждый коммутатор распределения с ядром. Однако проекты меньших территориальных локальных сетей редко используют концепцию коммутаторов ядра.

Ниже кратко описаны функции коммутаторов в территориальных сетях.

- **Уровень доступа к сети** предоставляет точки доступа, или подключения, устройств конечного пользователя. На этом уровне фреймы между двумя коммутаторами доступа к сети в нормальных условиях напрямую не передаются, а должны быть отправлены на уровень распределения.
- **Уровень распределения** объединяет каналы от коммутаторов уровня доступа к сети, пересылает фреймы между коммутаторами, но не служит для подключения устройств конечного пользователя.
- **Уровень ядра сети** объединяет каналы от коммутаторов уровня распределения в крупных территориальных локальных сетях и обеспечивает очень высокие скорости коммутации потоков данных.

Среда Ethernet и длина кабелей

Проектируя территориальную локальную сеть, инженер должен прежде всего оценить необходимую длину кабелей и выбрать на основе такой информации необходимую ему технологию Ethernet, поддерживающую сегменты нужной длины. Например, компания арендует помещения в пяти разных зданиях офисного комплекса, поэтому

инженеру нужно вычислить длину кабельного маршрута между зданиями и подобрать подходящий канал Ethernet.

Наиболее распространенными на сегодняшний день являются три технологии Ethernet: 10BASE-T, 100BASE-T, и 1000BASE-T. Для всех указанных технологий характерна максимальная длина кабеля в 100 м, но они слегка отличаются используемым кабелем. Ассоциации EIA/TIA стандартизировали и стандартизируют новые технологии Ethernet, а именно кабельные системы для них, в частности качество кабеля. В каждом из стандартов технологии Ethernet указана категория используемого кабеля, причем это минимально требуемая категория, в сети вполне может быть использован и более качественный кабель. Например, в стандарте 10BASE-T рекомендуется использовать кабель категории 3 (Category 3 — CAT3), в технологии 100BASE-T используется кабель пятой категории (CAT5), а в стандарте 1000BASE-T нужно использовать кабель категории 5e или 6 (CAT5e или CAT6). Если же инженер планирует использовать (чтобы не пропадала!) существующую кабельную систему, то он должен знать, какой тип кабеля UTP установлен, а также какие ограничения по скорости для соответствующей технологии Ethernet и кабеля существуют.

В нескольких технологиях Ethernet используются оптоволоконные кабели. В кабеле *неэкранированная витая пара* (Unshielded Twisted Pair — UTP) в качестве среды передачи используются медные провода, по которым протекает ток; в оптоволоконных кабелях используется ультратонкое стеклянное волокно, через которое передаются световые импульсы. Чтобы передать биты, оптические интерфейсы могут менять уровень света, делать ярче или слабее соответственно, кодируя таким образом 1 и 0 в кабеле.

Для оптических кабелей характерна большая длина кабельного сегмента, которая заметно превышает сотни метров для кабелей UTP в медных технологиях Ethernet. В оптических кабелях практически нет интерференции сигнала, вызываемой внешними источниками помех, и в зависимости от технологии в оптических коммутаторах в качестве источника сигнала могут быть использованы как лазеры, так и *светодиоды* (Light-Emitting Diode — LED). При использовании лазеров длина кабеля может быть очень большой, например, порядка 100 км на сегодняшний день. Для светодиодов, которые намного дешевле, характерны меньшие расстояния, вполне достаточные для подключения офисов и помещений в рамках территориальной сети.

И в заключение следует учесть, что используемая технология также определяет максимальную длину кабеля. Для оптоволоконных соединений следует помнить, что многомодовое оптическое волокно поддерживает меньшие расстояния, но зато стоит дешевле и позволяет использовать в качестве источника сигнала недорогие светодиоды. Второй тип оптоволоконных каналов — одномодовое волокно — поддерживает намного большую длину кабелей, но стоит дорого. Опять же коммутирующее оборудование на основе светодиодов (т.е. для многомодового оптоволокна) стоит значительно дешевле, чем оборудование с лазерными источниками (для одномодового оптоволокна).

В табл. 6.3 перечислены наиболее распространенные типы технологий Ethernet, их ограничения и типы кабелей.

Большинство инженеров обычно помнят только общие тенденции, а конкретные числа для максимальной длины кабеля подглядывают в каком-нибудь справочнике или таблице в книге (например, в табл. 6.3). Следует также принимать во вни-

мание физический путь, по которому будет проложен кабель, — он может значительно повлиять на длину кабельного сегмента. Например, нужно проложить кабель из одного конца здания в другой. "Напрямую" это может быть не так уж и много, но нужно учесть, что кабель будет проложен по коробам, т.е. сначала будут выведены спуски, например, из-под фальшпотолка до оборудования кабельного узла. Потом кабель будет проложен по коробам этажа, которые могут быть очень даже извилистыми и похожими на лабиринт, — такой маршрут, вполне очевидно, может быть не самым коротким. И только после расчета реальной протяженности кабельного маршрута следует обратиться к таблице или справочнику и выбрать подходящую среду передачи данных.

Таблица 6.3. Технологии Ethernet, их среда и максимальная длина сегмента (согласно стандартам IEEE)

Технология Ethernet	Среда	Максимальная длина сегмента, м (футы)
10BASE-T	Кабель UTP стандарта TIA категории CAT3 или лучше; используются две пары	100 (328)
100BASE-T	Кабель UTP стандарта TIA категории CAT5 или лучше; используются две пары	100 (328)
1000BASE-T	Кабель UTP стандарта TIA категории CAT5e или лучше; используются две пары	100 (328)
1000BASE-SX	Многомодовое оптоволокно	275 (853) для оптоволокна 62,5 микрона; 550 (1804,5) для оптоволокна 50 микрон
1000BASE-LX	Многомодовое оптоволокно	550 (1804,5) для оптоволокна 62,5 микрона и 50 микрон
1000BASE-LX	Одномодовое оптоволокно 9 микрон	5 км (3,1 мили)

Автопереговоры

Устройства Ethernet на концах канала связи должны использовать тот же стандарт, в противном случае они не смогут правильно отправлять данные. Например, сетевая плата не может использовать стандарт 100BASE-T, подразумевающий использование кабеля UTP с двумя витыми парами и скорости 100-Мбит/с, в то время как порт коммутатора на другом конце канала связи использует стандарт 1000BASE-T. Даже если использовать кабель для Gigabit Ethernet, канал связи не будет работать, если передача на одном конце осуществляется на скорости 100 Мбит/с, а прием на другом на скорости 1000 Мбит/с.

Переход на новые, более быстрые стандарты Ethernet становится проблемой, поскольку оба конца канала должны использовать тот же стандарт. Например, если заменить старый компьютер, возможно использовавший стандарт 100BASE-T, новым, то, вероятно, будет использован стандарт 1000BASE-T. Порт коммутатора на другом конце канала связи теперь должен использовать стандарт 1000BASE-T, поэтому потребуется модернизация коммутатора. Если все порты нового коммутатора

будут работать по стандарту 1000BASE-T, то придется обновить все остальные подключенные к нему компьютеры.

IEEE предлагает хорошее решение этой проблемы: автопереговоры IEEE. Автопереговоры IEEE — это стандарт IEEE 802.3u протокола, позволяющего двум узлам Ethernet на канале связи UTP вести переговоры о согласовании общей скорости и режима дуплексной передачи. Сообщения этого протокола передаются по кабелю UTP вне обычной рабочей частоты Ethernet, как внеполосные сигналы. Проще говоря, каждый узел заявляет о своих возможностях, а затем выбирает наилучшие из параметров, поддерживаемых обоими узлами: выбирается самая быстрая скорость, а дуплексный режим предпочтительней полудуплексного.

ВНИМАНИЕ!

Автопереговоры полагаются на тот факт, что, согласно стандартам IEEE 10BASE-T, 100BASE-T и 1000BASE-T, используется одинаковая схема расположения выводов. Достаточно добавить к ним еще две пары.

Многие сети используют автопереговоры каждый день, особенно между пользовательскими устройствами и уровнем доступа, как показано на рис. 6.13. Для поддержки Gigabit Ethernet компания проложила кабельную проводку с четырьмя парами проводов, поэтому они способны работать на скоростях 10, 100 и 1000 Мбит/с. Оба узла на каждом канале посылают друг другу сообщения автопереговоров. У коммутатора в данном случае все порты одинаковы, 10/100/1000, в то время как сетевые платы компьютеров поддерживают разные параметры.

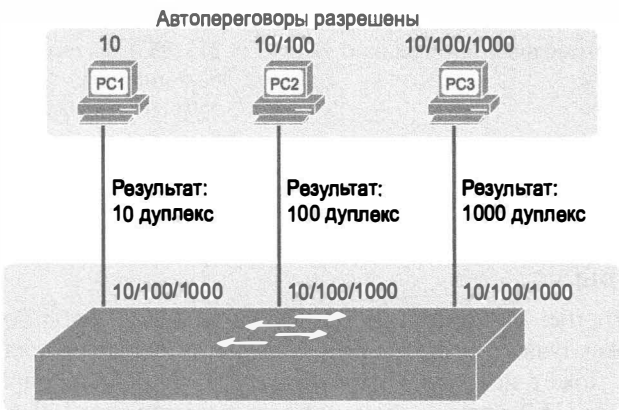


Рис. 6.13. Результаты автопереговоров IEEE с двумя правильно работающими узлами

Ниже приведена логика автопереговоров по каждому компьютеру.

- *Компьютер PC1.* Порт коммутатора утверждает, что может работать со скоростью 1000 Мбит/с, но сетевая плата компьютера PC1 требует наибольшей скорости 10 Мбит/с. Компьютер и коммутатор выбирают наивысшую скорость 10 Мбит/с и поддержку полного дуплекса.

- *Компьютер PC2.* Компьютер PC2 требует скорости 100 Мбит/с, значит, он может использовать стандарт 10BASE-T или 100BASE-T. Порт коммутатора и сетевая плата договариваются использовать наивысшую скорость 100 Мбит/с и полный дуплекс.
- *Компьютер PC3.* Компьютер использует сетевую плату 10/100/1000, поддерживающую все три скорости и стандарта, поэтому сетевая плата и порт коммутатора выбирают скорость 1000 Мбит/с и полный дуплекс.

Результаты автопереговоров, когда их использует только один узел

На рис. 6.13 представлены результаты автопереговоров IEEE, когда оба узла участвуют в процессе. Однако большинство устройств Ethernet способно отключить автопереговоры, поэтому столь важно знать, что происходит при попытке узла использовать автопереговоры, когда противоположный узел не дает ответа.

Отключение автопереговоров не всегда является плохой идеей, но их нужно либо разрешить на обоих концах канала связи, либо запретить также на обоих концах. Например, сетевые инженеры обычно отключают автопереговоры на каналах связи между коммутаторами и вручную задают скорость и дуплекс на обоих коммутаторах. Если автопереговоры разрешены на обоих концах канала связи, то узлы сами выберут наилучшую скорость и дуплекс. Однако, когда автопереговоры разрешены только на одном конце, может возникнуть множество проблем: канал связи может работать плохо или не работать вообще.

Автопереговоры IEEE определяют некоторые (стандартные) правила, которым должны следовать узлы при неудаче автопереговоров.

- *Скорость.* Используется наименьшая возможная скорость (обычно 10 Мбит/с).
- *Дуплекс.* При скорости 10 или 100 Мбит/с используется полудуплекс, в противном случае — полный дуплекс.

Коммутаторы Cisco дополняют базовую логику IEEE, поскольку они способны вычислить скорость другого узла даже без автопереговоров IEEE. В результате коммутаторы Cisco используют немного иную логику, когда автопереговоры терпят неудачу.

Стандартные действия коммутаторов Cisco при автопереговорах

- *Скорость.* Вычисляет скорость (без автопереговоров). При неудаче использует стандартное значение IEEE (наименьшая доступная скорость, обычно 10 Мбит/с).
- *Дуплекс.* Использует стандартное значения IEEE. При скорости 10 или 100 Мбит/с используется полудуплекс, в противном случае — полный дуплекс.

На рис. 6.14 представлены три примера, когда три пользователя изменили параметры своих сетевых плат и отключили автопереговоры. У коммутатора все порты одинаковы, 10/100/1000, с включенными автопереговорами. Сверху на рисунке представлены параметры конфигурации сетевых плат каждого компьютера, а рядом с каждым портом коммутатора отображен сделанный им выбор.

Рассмотрим каждый канал связи слева направо.

- *Компьютер PC1.* Коммутатор не получает сообщений автопереговоров, поэтому посылает электрический сигнал, запрашивающий у компьютера PC1 отправку данных на скорости 100 Мбит/с. Коммутатор использует стандартное значение дуплекса (согласно IEEE) для скорости 100 Мбит/с, т.е. полудуплекс.
- *Компьютер PC2.* Коммутатор использует те же этапы и логику, что и для канала связи с компьютером PC1, за исключением того, что коммутатор решил использовать полный дуплекс, поскольку скорость составляет 1000 Мбит/с.
- *Компьютер PC3.* Пользователь сделал плохой выбор: медленная скорость (10 Мбит/с) и худший режим — полудуплекс. Но коммутатор Cisco рассчитал скорость без автопереговоров IEEE, а затем использовал стандартное значение дуплекса IEEE для каналов связи на 10 Мбит/с, т.е. полудуплекс.

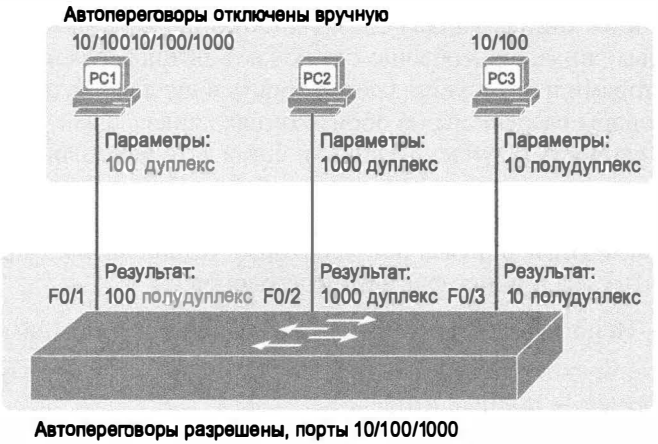


Рис. 6.14. Результаты автопереговоров IEEE при отключении автопереговоров на одной стороне

Случай с компьютером PC1 демонстрирует классический и, к сожалению, весьма обычный результат: *рассогласование дуплекса* (duplex mismatch). Оба этих узла (компьютер PC1 и порт F0/1 коммутатора SW1) используют скорость 100 Мбит/с, поэтому они могут передавать данные. Но компьютер PC1 использует полный дуплекс и, не пытаясь применить логику CSMA/CD, посылает фреймы в любое время. Порт F0/1 коммутатора использует полудуплекс и логику CSMA/CD. В результате, порт F0/1 будет обнаруживать коллизии на канале связи, хотя физически они и не происходят. Порт коммутатора будет регулярно прекращать передачу, ожидать и возобновлять передачу фреймов. В результате канал связи будет работать, но плохо.

Автопереговоры и концентраторы LAN

Концентраторы LAN также влияют на автопереговоры. Концентраторы в основном не реагируют на сообщения автопереговоров и не перенаправляют их. В результате соединенные через концентратор устройства должны использовать правила

IEEE, чтобы выбрать стандартные настройки, что зачастую приводит к использованию устройствами скорости 10 Мбит/с и полудуплексного режима.

На рис. 6.15 приведен пример небольшой локальной сети Ethernet, использующей концентратор 10BASE-T 20-летней давности. В этой сети все устройства и коммутатор имеют порты 10/100/1000. Концентратор поддерживает только стандарт 10BASE-T.

Рис. 6.15. Автопереговоры IEEE при концентраторе LAN

Обратите внимание, что устройствам справа требуется полудуплексный режим, поскольку концентратору следует использовать алгоритм CSMA/CD во избежание коллизий.

Обзор

Резюме

- Коммутаторы получают фреймы Ethernet на одном порту, а затем перенаправляют (коммутируют) их на один или несколько других портов.
- Коммутаторы LAN получают фреймы Ethernet, а затем принимают решение о коммутации: перенаправить фрейм на некий другой порт (порты) или проигнорировать его. Для этого они выполняют три действия:
  - на основании MAC-адреса устройства получателя принимается решение переслать фрейм или отфильтровать его (не пересылать);
  - на основании MAC-адресов устройств отправителей фреймов изучаются MAC-адреса и строится таблица коммутации;
  - за счет использования протокола распределенного связующего дерева (STP) поддерживается топология второго уровня без петель с другими коммутаторами.
- Чтобы принять решение о том, следует ли пересылать фрейм, коммутатор использует динамически создаваемую таблицу коммутации, в которой содержатся MAC-адреса и идентификаторы выходных интерфейсов. Чтобы принять решение о том, следует ли передать фрейм дальше или проигнорировать его, коммутатор сравнивает MAC-адрес получателя фрейма с записью в такой таблице.
- Коммутаторы строят таблицу адресов, просматривая входящие фреймы и выписывая из них MAC-адреса получателей. Если на вход какого-либо порта устройства получен фрейм и MAC-адрес в поле отправителя фрейма отсутствует в таблице коммутации, коммутатор создает соответствующую ему запись в таблице. В запись помещается адрес и идентификатор интерфейса, через который был получен фрейм.
- Когда в таблице нет соответствующей записи, коммутатор передает фрейм на все интерфейсы (кроме входящего), используя процесс лавинной рассылки.
- Третья главная функция коммутаторов локальных сетей заключается в предотвращении петлевых маршрутов с помощью протокола распределенного связующего дерева (STP). Без протокола STP фреймы могут бесконечно долго курсировать по петлевому маршруту, если в сети Ethernet есть резервные каналы. Чтобы избежать зацикливания фреймов, протокол STP блокирует некоторые порты, и они не могут пересылать данные. При этом в сети между сегментами (т.е. доменами коллизий) остается только один активный маршрут для передачи данных.
- Коммутация с буферизацией. Коммутатор получает фрейм полностью, до последнего бита, сохраняет, а затем начинает его передачу.
- Коммутация без буферизации. Коммутатор отправляет фрейм, как только будет получена нужная информация — MAC-адрес получателя. Этот метод

значительно уменьшает задержку, но фреймы с неправильной контрольной суммой (FCS) не будут отброшены устройством.

- Коммутация без фрагментации. Коммутатор начинает передачу, как только получит первые 64 байта фрейма. Этот метод позволяет исключить при коммутации большинство ошибочных фреймов и результатов коллизии.
- Первоначально термин *домен коллизий* относился к концепции всех портов Ethernet, передача фреймов которых приводила к конфликту с фреймами, передаваемыми другими устройствами в домене коллизий.
- Широковещательный домен Ethernet — это набор устройств, получающих данный широковещательный фрейм.
- По определению широковещательное сообщение, отправленное одним устройством в широковещательном домене, не пересылается устройствам в другом широковещательном домене.
- Во многих корпоративных сетях используются виртуальные локальные сети (VLAN). При использовании сетей VLAN коммутатор связывает свои интерфейсы с разными широковещательными доменами (сетями VLAN) и, согласно конфигурации, помещает одни интерфейсы в одну сеть VLAN, а другие — в другую. По существу, коммутатор создает несколько широковещательных доменов. Фактически все порты коммутатора формируют не единый широковещательный домен, а несколько, согласно конфигурации.
- Территориальные локальные сети охватывают крупные здания или несколько рядом расположенных зданий.
- Компания Cisco использует три термина для описания роли каждого коммутатора в проекте территориальной сети: уровень доступа к сети, уровень распределения и ядро сети.
  - *Коммутаторы доступа* подключены непосредственно к конечным пользователям, предоставляя им доступ к устройствам в локальной сети. Обычно коммутаторы доступа обеспечивают обмен трафиком с подключенными к ним устройствами конечного пользователя и находятся на краю локальной сети.
  - *Коммутаторы распределения* обеспечивают перенаправление трафика между коммутаторами доступа. В соответствии с проектом каждый из коммутаторов доступа соединен по крайней мере с одним коммутатором распределения, чтобы они могли перенаправить трафик другим частям LAN.
  - *Коммутаторы ядра* перенаправляют трафик между коммутаторами распределения.
- Автопереговоры IEEE — это стандарт IEEE 802.3u протокола, позволяющего двум узлам Ethernet на канале связи UTP вести переговоры о согласовании общей скорости и режима дуплексной передачи.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Какое из утверждений правильно описывает принимаемое коммутатором решение об отправке фрейма для известного ему одноадресатного (unicast) MAC-адреса получателя?

   А) Коммутатор сравнивает адрес получателя с таблицей коммутации (т.е. таблицей MAC-адресов).

   Б) Коммутатор сравнивает адрес отправителя с таблицей коммутации (т.е. таблицей MAC-адресов).

   В) Устройство рассылает фрейм через все интерфейсы в данной сети VLAN, кроме того, через который он был получен.

   Г) Коммутатор сравнивает IP-адрес получателя с MAC-адресом получателя.

   Д) Устройство сравнивает идентификатор входного интерфейса с MAC-адресом отправителя в таблице MAC-адресов.

2. Какое из утверждений правильно описывает принимаемое коммутатором решение об отправке фрейма для широковещательного (broadcast) MAC-адреса получателя?

   А) Коммутатор сравнивает адрес получателя с таблицей коммутации (т.е. таблицей MAC-адресов).

   Б) Коммутатор сравнивает адрес отправителя с таблицей коммутации (т.е. таблицей MAC-адресов).

   В) Устройство рассылает фрейм через все интерфейсы в данной сети VLAN, кроме того, через который он был получен.

   Г) Коммутатор сравнивает IP-адрес получателя с MAC-адресом получателя.

   Д) Устройство сравнивает идентификатор входного интерфейса с MAC-адресом отправителя в таблице MAC-адресов.

3. Какое из утверждений правильно описывает принимаемое коммутатором решение об отправке фрейма для неизвестного ему одноадресатного (unicast) MAC-адреса получателя?

   А) Устройство рассылает фрейм через все интерфейсы в данной сети VLAN, кроме того, через который он был получен.

   Б) Устройство пересылает фрейм через один интерфейс, для которого есть соответствующая запись в таблице MAC-адресов.

   В) Коммутатор сравнивает IP-адрес получателя с MAC-адресом получателя.

   Г) Устройство сравнивает идентификатор входного интерфейса с MAC-адресом отправителя в таблице MAC-адресов.

4. Какие действия выполняет коммутатор, если ему нужно принять решение о том, добавлять или нет новый MAC-адрес в таблицу MAC-адресов?

   А) Устройство сравнивает одноадресатный адрес получателя с записями в таблице коммутации (MAC-адресов).

   Б) Устройство сравнивает одноадресатный адрес отправителя с записями в таблице коммутации (MAC-адресов).

В) Устройство сравнивает идентификатор сети VLAN (ID) с записями в таблице коммутации (MAC-адресов).

Г) Устройство сравнивает IP-адрес получателя из записи в кеше ARP с записями в таблице коммутации (MAC-адресов).

5. Персональный компьютер PC1 с MAC-адресом 1111.1111.1111 подключен к интерфейсу Fa0/1 коммутатора SW1. Компьютер PC2 с MAC-адресом 2222.2222.2222 подключен к интерфейсу Fa0/2 коммутатора SW1, а компьютер PC3 с MAC-адресом 2222.2222.2222 подключен к интерфейсу Fa0/3 того же коммутатора. Изначально в таблице коммутатора нет никаких динамических записей о MAC-адресах. PC1 пересылает фрейм с адресом получателя 2222.2222.2222. Если после этого PC3 пересылает фрейм компьютеру PC2 с адресом получателя 2222.2222.2222, что будет происходить в коммутаторе? (Выберите два ответа.)

А) Коммутатор перешлет фрейм через интерфейс Fa0/1.

Б) Коммутатор перешлет фрейм через интерфейс Fa0/2.

В) Коммутатор перешлет фрейм через интерфейс Fa0/3.

Г) Коммутатор отбросит (или отфильтрует) такой фрейм.

6. В каком случае два компьютера будут в том же домене коллизий?

А) Если они разделены концентратором Ethernet.

Б) Если они разделены прозрачным мостом.

В) Если они разделены коммутатором Ethernet.

Г) Если они разделены маршрутизатором.

7. В каком случае два компьютера будут в том же широковещательном домене? (Выберите три ответа.)

А) Если они разделены концентратором Ethernet.

Б) Если они разделены прозрачным мостом.

В) Если они разделены коммутатором Ethernet.

Г) Если они разделены маршрутизатором.

8. В каком из стандартов Ethernet максимальная длина кабеля не может превышать 100 м? (Выберите два ответа.)

А) 100BASE-T.

Б) 1000BASE-LX.

В) 1000BASE-T.

Г) 100BASE-FX.

9. Коммутатор Cisco локальной сети непосредственно подключен к трем компьютерам (PC1, PC2 и PC3) кабелем, поддерживающим UTP Ethernet со скоростью 1000 Мбит/с (1 Гбит/с). Компьютер PC1 использует сетевую плату, поддерживающую только стандарт 10BASE-T, компьютер PC2 имеет сетевую плату 10/100, а PC3 — 10/100/1000. С учетом, что компьютеры и коммутатор используют автопереговоры IEEE, какие компьютеры будут использовать полудуплексный режим?

А) PC1.

Б) PC2.

В) PC3.

Г) Ни один из компьютеров не будет использовать полудуплекс.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 6.4.

Таблица 6.4. Ключевые темы главы 6

Элемент	Описание	Страница
Список	Действия, выполняемые коммутаторами	191
Рис. 6.4	Пример коммутации и фильтрации фреймов	192
Рис. 6.5	Решение о перенаправлении при двух коммутаторах	193
Рис. 6.6	Самообучение коммутатора: пустая таблица и добавление двух записей	194
Табл. 6.1	Методы коммутации фреймов	197
Список	Преимущества коммутации	197
Список	Алгоритмы коммутации, фильтрации фреймов и построения таблиц MAC-адресов	198
Список	Определения широковещательного домена и домена коллизий	200
Табл. 6.2	Преимущества сегментации сетей Ethernet с помощью концентраторов, коммутаторов и маршрутизаторов	202
Рис. 6.11	Сеть с двумя сетями VLAN и одним коммутатором	203
Список	Стандартные действия коммутаторов Cisco при автопереговорах	209

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

автопереговоры (autonegotiation), широковещательный домен (broadcast domain), широковещательный фрейм (broadcast frame), домен коллизий (collision domain), коммутация без буферизации пакетов (cut-through switching), лавинная рассылка (flooding), коммутация без фрагментации (fragment-free switching), протокол распределенного связующего дерева (Spanning Tree Protocol — STP), коммутация с буферизацией пакетов (store-and-forward switching), одноадресатный фрейм с неизвестным получателем (unknown unicast frame), виртуальная локальная сеть (Virtual LAN — VLAN)

Ответы на контрольные вопросы:

1 Б. **2** В. **3** А. **4** Б. **5** А и Б. **6** А. **7** А, Б, и В. **8** Б и Г. 9 Г.

Глава 7

Работа с коммутаторами Cisco

После покупки коммутатора Catalyst компании Cisco, когда вы достанете его из коробки, подключите питание от настенной розетки, подключите к нему рабочие станции правильным кабелем UTP, и устройство сразу же заработает. Никаких дополнительных действий можно не предпринимать, настроек не вводить и, тем более, не указывать коммутатору, как передавать фреймы Ethernet. Стандартные настройки коммутатора предполагают, что все интерфейсы устройства включены, что использованы правильные кабели и к коммутатору подключены правильные устройства, поэтому он может передавать и принимать фреймы с данными.

Тем не менее в большинстве сетей администраторы хотят иметь возможность проверить состояние коммутатора, просмотреть информацию о том, чем занимается устройство в данный момент, и, возможно, настроить некоторые дополнительные функции. Зачастую также необходимо включить определенные функции безопасности, чтобы позволить сетевым инженерам безопасно подключаться к устройству и контролировать поведение пользователей в сети. Чтобы выполнять все вышеперечисленные действия, у сетевого инженера должна быть возможность подключиться к интерфейсу командной строки коммутатора.

В этой главе рассказывается о том, как получить доступ к интерфейсу командной строки коммутаторов компании Cisco, какие команды использовать, чтобы проверить текущую работу устройства, и как осуществить некоторые базовые настройки устройства. В текущей главе основной упор сделан на процессы конфигурации, а не на какие-либо определенные наборы команд. В части II более подробно описаны различные команды для настройки устройства с помощью интерфейса командной строки.

В этой главе рассматриваются следующие экзаменационные темы

Технологии коммутации сетей LAN

Настройка и проверка начальной конфигурации коммутатора, включая управление удаленным доступом.

Команды операционной системы Cisco IOS для базовой настройки коммутатора.

Проверка состояния сети и работоспособности коммутатора с помощью базовых сетевых утилит ping, telnet и ssh.

Основные темы

Доступ к интерфейсу командной строки коммутатора Cisco Catalyst 2960

В маршрутизаторах и большинстве моделей коммутаторов Catalyst компании Cisco используется концепция *интерфейса командной строки* (Command-Line Interface — CLI). CLI представляет собой текстовый интерфейс, в котором пользователь, обычно сетевой инженер, вводит некоторые команды в виде текста. Нажатием клавиши <Enter> такая команда передается коммутатору и указывает устройству, что нужно сделать. Коммутатор выполняет действие, указанное в команде, и в определенных случаях выдает в ответ некоторое информационное сообщение, содержащее результат выполнения команды.

Коммутаторы Cisco Catalyst позволяют также контролировать и настраивать себя. Например, коммутатор может предоставлять веб-интерфейс, чтобы сетевой инженер мог использовать веб-браузер для доступа к веб-серверу, выполняющемуся на коммутаторе. Коммутаторы можно также контролировать при помощи программного обеспечения управления сетью, обсуждаемого во втором томе книги.

В этой книге обсуждаются только коммутаторы Cisco Catalyst корпоративного класса, а также использование CLI Cisco для их контроля. Данный раздел начинается с более подробного обсуждения коммутаторов Cisco Catalyst, а затем переходит к тому, как сетевой инженер может получить доступ к интерфейсу CLI, чтобы вводить команды.

Коммутаторы Cisco Catalyst и модель 2960

Компания Cisco производит широкий ассортимент коммутирующих устройств Catalyst для локальных сетей, модели которых объединены в *серии* (series), или *семейства* (families), устройств. Каждая серия устройств включает в себя несколько специфических моделей коммутаторов с похожим набором функций, примерно одинаковым соотношением "цена–производительность" и близкими внутренними компонентами устройств.

Компания Cisco позиционирует коммутаторы серии (или семейства) 2960 как недорогое полнофункциональное устройство для *кабельных узлов* (wiring closet) корпоративного уровня. Такое утверждение означает, что эти коммутаторы рекомендуется использовать в качестве устройств доступа к сети, как показано на рис. 6.12 главы 6.

На рис. 7.1 представлены фотографии коммутаторов разных моделей серии 2960. Например, пятый коммутатор имеет 48 портов для разъемов RJ-45 кабеля UTP со скоростями передачи данных 10/100, следовательно, такие порты могут автоматически согласовывать режим работы 10BASE-T или 100BASE-T Ethernet. Справа у этих коммутаторов есть также несколько интерфейсов 10/100/1000, предназначенных для подключения к ядру корпоративной территориальной сети LAN.

В документации и книгах по устройствам компании Cisco физические соединения коммутатора называют *интерфейсами* (interface), или *портами* (port). Каждый интерфейс обозначается номером в формате x/y, где x и y — разные числа. В коммутаторе модели 2960 число перед косой чертой (/) всегда равно 0, следовательно, первый ин-

терфейс со скоростью 10/100 обозначается как 0/1, второй — 0/2 и т.д. У каждого интерфейса также есть название, связанное с его технологией работы, например, полное название интерфейса с номером выглядит как "interface FastEthernet 0/1", означающее первый интерфейс 10/100. Любой интерфейс, поддерживающий гигабитовую скорость работы, обозначается как "GigabitEthernet", например, первый интерфейс со скоростями 10/100/1000 будет обозначаться в интерфейсе командной строки как "interface gigabitethernet 0/1".

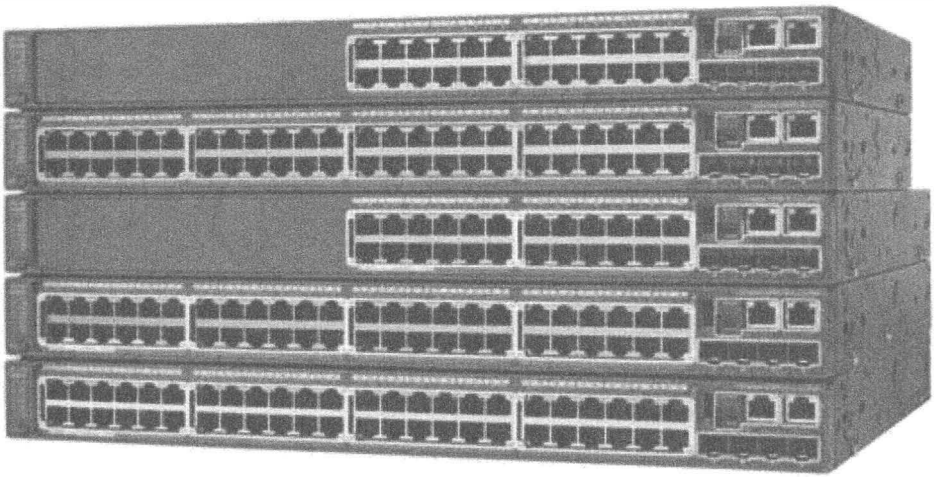


Рис. 7.1. Коммутаторы серии Cisco 2960 Catalyst

Световые индикаторы коммутатора

Представим себе ситуацию, когда сетевому инженеру нужно быстро проверить, как работает коммутатор, определить состояние портов, чтобы найти и устранить какую-либо проблему в сети. В таком случае очень много времени будет потрачено на консольное подключение к устройству, ввод и интерпретацию результатов команд в интерфейсе командной строки операционной системы Cisco IOS. Чтобы упростить такую задачу, в коммутаторах компании Cisco используются *светодиодные индикаторы* (LED), позволяющие получить некоторую информацию о состоянии устройства, причем как в процессе загрузки устройства, так и в ходе его нормальной обычной работы. Прежде чем переходить к обсуждению интерфейса командной строки (CLI), кратко опишем, какие светодиодные индикаторы есть у коммутатора и что они означают.

У большинства коммутаторов Catalyst компании Cisco на передней панели есть светодиодные индикаторы, в том числе и индикаторы для каждого интерфейса Ethernet. На рис. 7.2 показан коммутатор серии 2960, у которого есть пять индикаторов слева на передней панели, светодиодный индикатор для каждого порта и *кнопка переключения режима* (mode button).

На рис. 7.2 показан внешний вид передней панели устройства, а в табл. 7.1 приведено их описание.

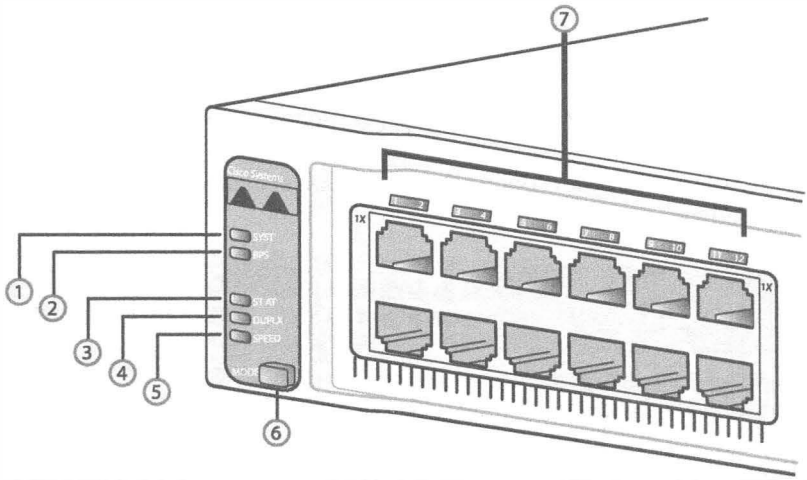


Рис. 7.2. Индикаторы коммутатора 2960 и кнопка переключения режимов

Таблица 7.1. Светодиодные индикаторы и кнопка коммутатора, показанные на рис. 7.2

Номер на рис. 7.2	Название	Описание
1	SYST (system, системный)	Общее состояние системы
2	RPS (Redundant Power Supply, резервный блок питания)	Показывает состояние дополнительного (резервного) блока питания
3	STAT (status, состояние)	Будучи включенным (светится зеленым), сигнализирует о нормальном состоянии порта
4	DUPLX (duplex, дуплексный)	Будучи включенным (светится зеленым), свидетельствует о работе порта в дуплексном режиме, выключенным — в полудуплексном
5	SPEED	Будучи выключенным, свидетельствует о скорости работы 10 Мбит/с, включенным (светится зеленым) — 100 Мбит/с, мигающим зеленым — 1 Гбит/с
6	MODE	Кнопка переключения режимов индикаторов (STAT, DUPLX, SPEED)
7	Port	Индикаторы указывают разные состояния в зависимости от режима, выбранного кнопкой MODE

Рассмотрим несколько примеров, чтобы понять, что означают различные индикаторы. Например, разберемся, о чем сигнализирует индикатор SYST в процессе работы устройства. Для коммутатора серии 2960 этот индикатор может работать в трех режимах:

- не светится — питание коммутатора не включено;
- светится зеленым — питание устройства включено и коммутатор работает нормально (операционная система Cisco IOS загрузилась);
- светится оранжевым — питание включено, но операционная система работает не правильно.

Если бросить быстрый взгляд на индикатор SYST, то сразу же становится понятным, работает устройство или нет. Если не работает, то можно грубо определить причину: нет электропитания (индикатор не светится) или операционная система Cisco IOS не загружена (индикатор светится оранжевым) и отказал какой-то аппаратный модуль. Таким образом, сначала нужно проверить и переключить питание; если симптомы не изменились, скорее всего, придется обращаться *в службу технической поддержки компании Cisco* (Cisco Technical Assistance Center — TAC) либо поставщика оборудования.

Кроме системного индикатора устройства, функции которого достаточно очевидны, у коммутаторов обычно есть индикаторы для каждого порта, размещенные сверху или снизу разъема интерфейса. Смысл их показаний зависит от режима работы, выбранного кнопкой переключения режимов (см. рис. 7.2). Нажимая такую кнопку, можно циклически переключать режим работы индикаторов между состояниями STAT, DUPLX и SPEED (см. табл. 7.1 и рис. 7.2). Чтобы перейти в нужный режим, следует нажать кнопку переключения режимов один или два раза.

Например, в режиме STAT (status — состояние) для каждого порта отображается его состояние:

- выключен — канал связи не работает (или отключен);
- светится зеленым — канал связи работает, но через интерфейс не передаются данные;
- мигает зеленым — канал связи работает, через интерфейс передаются данные;
- мигает оранжевым — порт административно выключен или блокирован.

Для сравнения: если переключить индикаторы в режим SPEED (скорость порта), то они будут просто показывать рабочую скорость передачи данных для интерфейса: выключенный индикатор свидетельствует о скорости 10 Мбит/с, светящийся зеленым — 100 Мбит/с, а мигающий зеленым — 1000 Мбит/с (т.е. 1 Гбит/с).

В действительности значение индикаторов и доступные режимы работы для разных серий устройств могут быть существенно разными и совпадать только для устройств одной серии. Поэтому запоминать на память значения определенных комбинаций индикаторов и режимы работы, скорее всего, бесполезно. Не будем подробно останавливаться на всех возможных вариантах работы индикаторов для каждой модели коммутатора или даже для целых семейств устройств, но если читателю нужна дополнительная информация, то он может обратиться к документации компании Cisco. Тем не менее важно запомнить основную идею, т.е. для чего предназначены индикаторы, а также помнить о том, что кнопка переключения режимов используется для изменения класса отображаемой информации. Следует также помнить, для чего нужен индикатор SYST и что он показывает.

В подавляющем большинстве случаев коммутатор включается, успешно загружает операционную систему Cisco IOS, и инженер получает доступ к интерфейсу командной строки устройства, чтобы управлять устройством и контролировать его. В разделе ниже основное внимание будет уделено тому, как получить доступ к интерфейсу командной строки.

Доступ к интерфейсу командной строки системы IOS

Как и любые другие аппаратные средства, коммутаторы Cisco нуждаются в программном обеспечении — операционной системе. Компания Cisco использует операционную систему IOS (Internetwork Operating System — *межсетевая операционная система*).

Программное обеспечение IOS компании Cisco для коммутаторов Catalyst реализует алгоритмы обработки потоков данных и управляет функциями коммутирующего устройства. Операционная система не только контролирует производительность, функции и поведение коммутатора, но и предоставляет дружественный пользователю *интерфейс командной строки*, обычно обозначаемый аббревиатурой CLI (Command Line Interface). Интерфейс CLI операционной системы Cisco IOS подразумевает использование какой-либо программы эмуляции терминала, которая позволяет передавать вводимый в ней текст устройству. Когда пользователь нажимает клавишу <Enter>, программа пересылает текст, а коммутатор обрабатывает его так, как если бы это была команда, и возвращает некоторый ответ обратно программе-эмулятору.

Получить доступ к интерфейсу командной строки можно с помощью трех популярных методов: через консольное подключение, через протокол Telnet и различные варианты программы протокола SSH (Secure Shell). Два последних метода предполагают, что коммутатор установлен в уже работающей, причем правильно, сети IP, через которую осуществляется дистанционное управление устройством. Консольное подключение — это специализированный физический порт устройства для доступа к интерфейсу командной строки и настройки. На рис. 7.3 показаны различные варианты доступа к интерфейсу командной строки коммутатора.

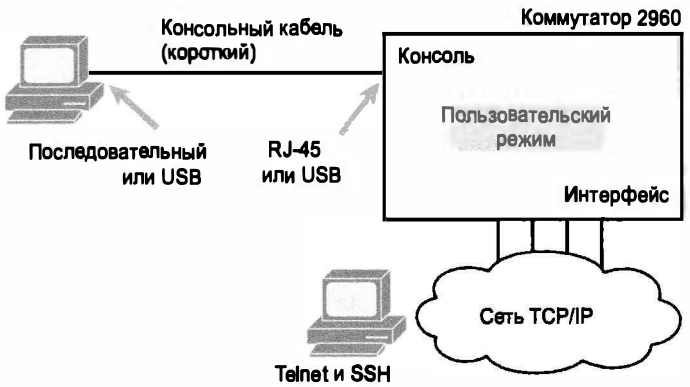


Рис. 7.3. Доступ к интерфейсу командной строки

Для консольного доступа требуется наличие физического соединения между компьютером (или другим пользовательским устройством) и консольным портом коммутатора, а также наличие на компьютере соответствующего программного обеспечения. Протоколы Telnet и SSH также требуют наличия соответствующего программного обеспечения на устройстве пользователя, но для передачи данных они полагаются на существующую сеть TCP/IP. Далее подробно описано, как подключить консоль и установить программное обеспечение для каждого метода, чтобы получить доступ к интерфейсу командной строки (CLI).

Кабельная проводка консольного соединения

Физическое консольное соединение, и прежнее, и новое, использует три основных компонента: физический консольный порт на коммутаторе, физический последовательный порт на компьютере и кабель, соединяющий консольный и последовательный порты. Однако физические детали кабельной проводки со временем немного изменились, главным образом из-за усовершенствований и изменений в аппаратных средствах компьютера.

Прежние консольные соединения использовали последовательный порт компьютера, консольный кабель и разъем RJ-45 на коммутаторе. Последовательный порт компьютера обычно имеет разъем D-shell (почти прямоугольный) с девятью контактами (зачастую называемый DB-9). Прежние коммутаторы, а также некоторые нынешние модели используют для консольного порта разъем RJ-45. Такое кабельное соединение показано на рис. 7.4, *слева*.

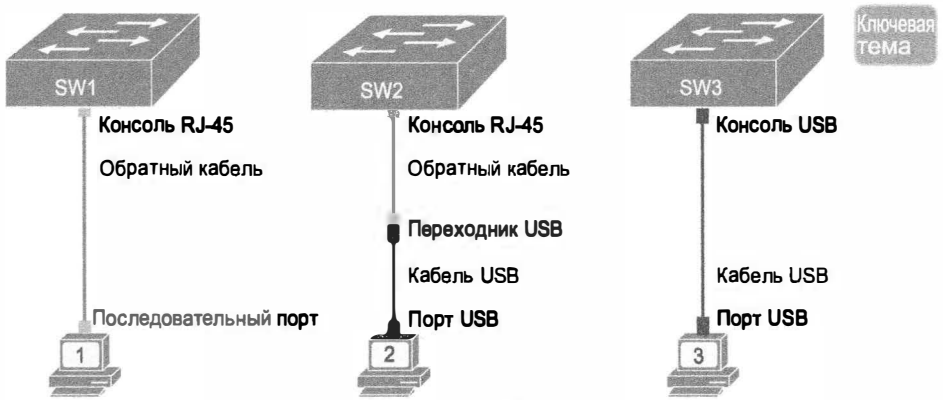


Рис. 7.4. Возможности кабельного подключения к консоли

Можно использовать специализированный консольный кабель (связывающий новые коммутаторы и маршрутизаторы Cisco) или проложить собственный консольный кабель, используя кабель UTP и стандартный переходник разъемов RJ-45 – DB-9. Такой переходник можно купить в большинстве компьютерных магазинов. Для кабеля UTP здесь используется обратная схема расположения выводов, в отличие от стандартных кабелей Ethernet. Кабель использует восемь проводов, контакт 1 перекрешен с контактом 8, контакт 2 с контактом 7, контакт 3 с контактом 6 и т.д.

Для последовательной передачи данных на современных компьютерах вместо последовательных портов все чаще используются порты USB (Universal Serial Bus — *универсальная последовательная шина*). Компания Cisco также начала выпускать более новые маршрутизаторы и коммутаторы с портами USB для консольного доступа. В самом простом случае можно использовать любой порт USB на компьютере и кабель USB, подключенный к консольному порту USB на коммутаторе или маршрутизаторе, как показано на рис. 7.4, *справа*.

Еще одна возможность представлена на рис. 7.4, *посередине*. У многих компьютеров больше нет последовательных портов, но у многих существующих маршрутизаторов Cisco и коммутаторов есть только консольный порт RJ-45 и нет консольного порта USB. Для подключения такого компьютера к консольному порту маршрути-

затора или коммутатора необходим переходник между устаревшим консольным кабелем и разъемом USB (см. рис. 7.4, *посередине*).

ВНИМАНИЕ!

При использовании портов USB, как правило, приходится устанавливать программный драйвер, чтобы операционная система компьютера знала, что устройство на другом конце соединения USB является консолью устройства Cisco.

Настройка эмулятора терминала для консоли

После физического подключения компьютера к консольному порту, на компьютере следует установить и настроить пакет программ эмулятора терминала. Эмулятор рассматривает все данные как текст. Он принимает текст, введенный пользователем, и посылает его по консольному соединению на коммутатор. Точно так же все биты, поступающие на компьютер по консольному соединению, отображаются как текст для удобства чтения пользователем.

Эмулятор терминала нужно настроить так, чтобы он использовал последовательный порт компьютера, а настройки его должны совпадать со стандартными настройками коммутатора. Стандартные настройки консольного порта коммутатора должны быть следующими:

Ключевая тема **Стандартные настройки консольного порта коммутатора компании Cisco**

- скорость 9600 бит/с;
- без аппаратного контроля потока (hardware flow control);
- 8 бит данных;
- без контроля четности (no parity);
- 1 стоповый бит.

Последние три настройки по первым буквам параметров для простого запоминания записывают как "8N1".

Один из эмуляторов терминала, Zterm Pro, показан на рис. 7.5. На заднем плане находится окно, созданное программным обеспечением эмулятора, с выводом команды `show`. На переднем плане (вверху слева) представлено окно параметров, демонстрирующее перечисленные выше стандартные настройки консоли.

Доступ к интерфейсу командной строки с помощью протоколов Telnet и SSH

Приложение Telnet стандартного стека протоколов TCP/IP позволяет эмулятору терминала взаимодействовать с устройством и выглядит очень похоже на консольное подключение. В отличие от последнего, для отправки и получения данных в этом приложении используется сеть IP, а не специализированный кабель и физический порт устройства. В протоколе уровня приложений Telnet программа эмуляции терминала рабочей станции называется *клиентом Telnet*, а устройство, принимающее команды и отвечающее на них, — *сервером Telnet*. Telnet — это основанный на механизме TCP протокол уровня приложений, использующий стандартный зарезервированный порт с номером 23.

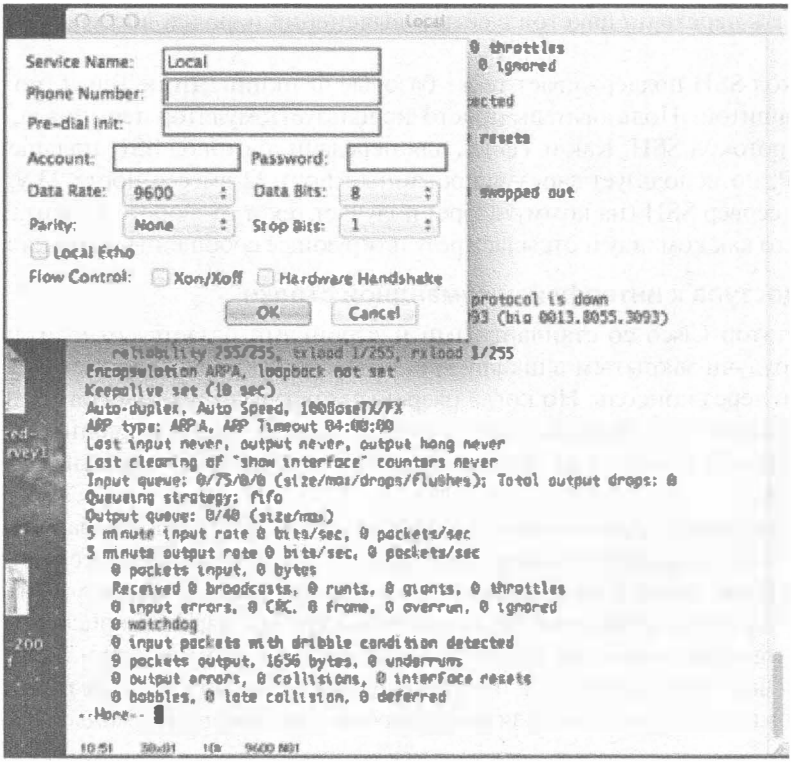


Рис. 7.5. Настройки программы эмуляции терминала при подключении к консоли

Чтобы использовать службу Telnet, пользователь должен установить клиент Telnet на своей рабочей станции. (Как было сказано выше, большинство современных пакетов программного обеспечения эмуляции терминала включает в себя клиент Telnet и SSH.) Сервер Telnet на коммутаторе запущен стандартно и работает всегда, тем не менее, чтобы он мог отправлять и принимать пакеты IP, для устройства нужно задать IP-адрес. (Установка IP-адреса подробно описана в главе 8.) Кроме того, чтобы рабочая станция сетевого инженера или администратора и коммутатор могли обмениваться пакетами, сеть между двумя устройствами должна правильно работать.

Многие из сетевых инженеров постоянно используют клиент Telnet, чтобы контролировать и настраивать коммутаторы, — инженер может находиться на своем рабочем месте, ему не нужно идти в соседнее здание, а иногда и ехать в другую область или в другую страну, тем не менее у него есть возможность что-либо выполнять в интерфейсе командной строки устройств. Протокол Telnet пересылает все данные (в том числе имя пользователя и пароль для доступа к устройству) в виде открытого текста, что представляет собой очень серьезную потенциальную брешь в системе безопасности.

Хотя технология Telnet работает вполне хорошо, многие сетевые инженеры используют вместо него *протокол Secure Shell* (SSH), позволяющий преодолеть проблему защиты, присущую технологии Telnet. Telnet передает на коммутатор все данные (включая имена пользователей и пароли) как открытые текстовые данные. Протокол SSH шифрует содержимое сообщений, включая пароли, предотвращая

возможность перехвата пакетов в сети и выяснения паролей доступа к сетевым устройствам.

Протокол SSH поддерживает те же базовые функции, что и Telnet, но с дополнительной защитой. Пользователь просто использует эмулятор терминала, поддерживающий протокол SSH. Как и Telnet, для передачи протокол SSH полагается на протокол TCP, но использует зарезервированный порт 22 вместо порта 23 у Telnet. Как и у Telnet, сервер SSH (на коммутаторе) получает текст от любого клиента SSH, обрабатывает его как команду и отсылает результирующее сообщение назад клиенту.

Пароли доступа к интерфейсу командной строки

Коммутатор Cisco со стандартными настройками остается относительно защищенным, будучи закрытым в шкафу кабельного узла, поскольку допускает управление только через консоль. Но когда разрешен доступ по технологиям Telnet и (или) SSH, необходимо установить пароль, чтобы только уполномоченные лица имели доступ к его интерфейсу CLI. Кроме того, для безопасности защитить паролем стоит и консоль.

Чтобы настроить на коммутаторе простую аутентификацию по паролю, для пользователей Telnet достаточно ввести лишь несколько команд. Процесс настройки описан в этой главе ниже, а необходимые команды показаны в крайнем правом столбце табл. 7.2. В таблице приведены две команды настройки паролей консоли и терминала vty; после того как указанные команды будут введены, коммутатор начинает при подключении выдавать запрос на простую аутентификацию (это результат команды `login`) и ожидает от пользователя ввода пароля, указанного в команде `password`.

Таблица 7.2. Настройка паролей доступа для интерфейса командной строки консоли и подключений Telnet

Метод доступа	Тип пароля	Пример конфигурации

Консоль	Пароль консоли	`line console 0`
		`login`
		`password faith`

Приложение Telnet	Пароль терминала vty	`line vty 0 15`
		`login`
		`password love`

В коммутаторах компании Cisco консольный порт обозначается как специализированная линия, а именно как консольная линия 0. Устройство поддерживает также 16 одновременных сеансов протокола Telnet, называемых *виртуальными линиями* (vty) и нумеруемых от 0 до 15. Команда `line vty 0 15` указывает коммутатору, что следующие за ней настройки будут применены ко всем 16 возможным виртуальным терминальным соединениям с коммутатором (от 0 до 15), при этом такие настройки будут использоваться как для сеансов Telnet, так и SSH.

После ввода конфигурационных команд, перечисленных в табл. 7.2, у каждого пользователя, подключающегося с помощью консоли к устройству, будет запрошен пароль, и он должен будет ввести слово **`faith`**, согласно настройкам. Для каждого нового сеанса Telnet также будет запрашиваться пароль, в данном случае слово `love`. В рассмотренной конфигурации не нужно вводить имя пользователя, только пароль.

Чтобы настроить доступ SSH к устройству, понадобится добавить еще несколько команд, кроме тех, которые указаны в табл. 7.2. Протокол требует наличия открытого ключа шифрования для обмена общим ключом между сервером и клиентами, который, собственно, и будет использоваться для шифрования трафика. Кроме того, протокол SSH требует более высокого уровня безопасности, поэтому в нем запрашивается как имя пользователя, так и пароль. Полезные примеры, этапы и настройка нужных характеристик и параметров протокола SSH подробнее рассмотрены в главе 8.

Пользовательский и привилегированный режимы

Три рассмотренных выше метода доступа к устройству (консольное подключение, сеанс Telnet и SSH) подключают пользователя к интерфейсу командной строки в *пользовательском режиме* (user mode), или *режиме EXEC обычного пользователя* (user EXEC mode). В этом режиме пользователь преимущественно может просматривать различную информацию, но не настроить или "сломать" что-то. Приставка "EXEC" подчеркивает, что когда пользователь вводит какую-либо команду, коммутатор ее выполняет и выводит некоторое сообщение о результатах работы.

Операционная система Cisco IOS поддерживает режим EXEC и с большими возможностями — *привилегированный режим* (privileged mode), он же *режим enable* (enable mode), или *привилегированный режим EXEC* (privileged EXEC mode). Привилегированный режим получил свое имя от команды `enable`, переводящей пользователя из пользовательского режима в привилегированный, как показано на рис. 7.6. Другое название этого режима, privileged, следствие того факта, что он допускает выполнение команд, обладающих привилегией вносить изменения. Например, команда `reload`, позволяющая перезагрузить коммутатор Cisco IOS, может быть отдана только в привилегированном режиме.

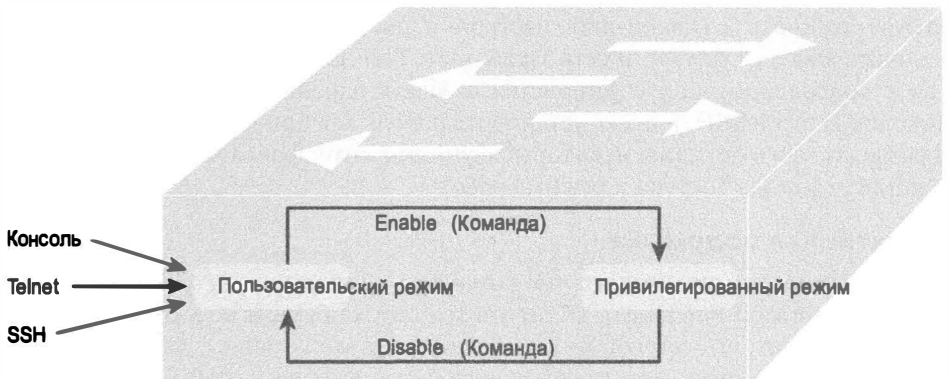


Рис. 7.6. Пользовательский и привилегированный режимы

ВНИМАНИЕ!

Следует запомнить, что если в приглашении командной строки в конце имени устройства стоит угловая скобка (>), то это обычный пользовательский режим; если в конце имени устройства стоит символ решетки (#), то это привилегированный режим.

Пример 7.1. Переход между различными режимами EXEC на коммутаторе Certskills1

```
Press RETURN to get started.

User Access Verification

Password:
Certskills1>
Certskills1> reload
Translating "reload"
% Unknown command or computer name, or unable to find computer address
Certskills1> enable
Password:
Certskills1#
Certskills1# reload
Proceed with reload? [confirm] y
00:08:42: %SYS-5-RELOAD: Reload requested by console. Reload Reason:
Reload Command.
```

ВНИМАНИЕ!

Зачастую команды, которые могут быть использованы как в обычном режиме EXEC, так и в привилегированном режиме EXEC, называют просто командами EXEC.

Этот пример первым в книге демонстрирует вывод команд CLI, поэтому следует обратить внимание на некоторые соглашения. Полужирным шрифтом выделен текст, введенный пользователем, а обычным — посланный коммутатором назад эмулятору терминала. Кроме того, пароли обычно не отображаются на экране для безопасности. И наконец, как уже упоминалось, этот коммутатор носит имя хоста `Certskills1`, поэтому приглашение к вводу команд слева отображает это имя хоста в каждой строке.

Выше были описаны основные особенности стандартной настройки коммутаторов Cisco, которые следует учитывать при покупке и установке нового устройства. Коммутатор будет работать без какой-либо настройки, достаточно просто включить питание, подключить кабели Ethernet, и сеть заработает. Тем не менее рекомендуется подключиться к консольному порту устройства и задать пароли для консольного доступа, протоколов Telnet и SSH, а также установить пароль для привилегированного режима.

В разделах ниже описаны некоторые функции интерфейса командной строки, не зависящие от метода доступа и режима работы.

Интерактивная подсказка

Если распечатать справочник по командам операционной системы Cisco IOS (Cisco IOS Command Reference), то получится стопка бумаги не меньше метра высотой. Никто не ждет, что сетевой инженер запомнит все команды, да и никто этого не сможет. Существует несколько простых и удобных методов запоминания команд операционной системы, а также средств, помогающих сэкономить время на печать команд. По мере продвижения ко все более сложным экзаменам и сертификациям компании Cisco количество команд, которые нужно запомнить, будет расти, поэтому следует знать, как пользоваться подсказкой, встроенной в операционную систему.

В табл. 7.3 перечислены пункты интерактивной подсказки операционной системы IOS, связанные с вызовом команд, вводившихся ранее инженером. Обратите внимание на то, что в первом столбце таблицы слово *команда* — это какая угодно произ-

вольная команда, а не команда с именем *команда*; аналогично слово *параметр* — это любой параметр команды. Например, в третьей строке таблицы приведен пример интерактивной подсказки в виде *команда* ?, который может означать как команду `show ?`, так и команду `copy ?` или любую другую команду со знаком вопроса в конце, в результате которой будет выведена подсказка по доступным параметрам.

Таблица 7.3. Интерактивная подсказка операционной системы Cisco IOS

Вводимая команда	Подсказка
?	Список всех команд, доступных для текущего режима
help	Текст, описывающий, как пользоваться подсказкой. Подсказка для команд не выводится
команда ?	Текстовая подсказка, описывающая первый параметр команды
com?	Эта команда выдаст список команд, начинающихся с символов com
команда начальные_буквы_параметра?	Список параметров, начинающихся с указанной последовательности символов (обратите внимание: между параметром и знаком вопроса нет пробела!)
команда начальные_буквы_параметра<Tab>	Если нажать клавишу <Tab> в середине любого параметра какой-либо команды, интерфейс командной строки или закончит слово команды, или ничего не сделает. Если нажатие клавиши не приводит к какому-либо результату, значит, у команды есть несколько параметров, начинающихся с уже введенной последовательности символов, и интерфейс не знает, какой из них выбрать
команда параметр1 ?	Если между параметром команды и знаком вопроса стоит пробел, интерфейс командной строки выводит список следующего параметра команды и короткое текстовое описание каждого

Следует помнить, что интерфейс командной строки операционной системы Cisco IOS реагирует на ввод знака вопроса (?) мгновенно — нажимать клавишу <Enter> или какую-либо другую не нужно. Операционная система после вывода подсказки повторно выводит все, что было введено в командной строке до символа ?, чтобы облегчить жизнь системному инженеру. Если нажать клавишу <Enter> сразу же после ввода знака вопроса, система Cisco IOS повторит команду без знака вопроса в конце и выполнит ее по нажатию клавиши, т.е. только с теми параметрами, которые были введены до знака вопроса.

Вид подсказки зависит от текущего режима интерфейса командной строки. Например, если нажать <?> в режиме обычного (непривилегированного) пользователя, будут показаны команды, которые разрешено выполнять из этого режима, а команды из привилегированного режима отображаться не будут. Иначе будет также выглядеть интерактивная подсказка при использовании ее в режиме конфигурации — как минимум, она будет показывать совсем другие команды. Подробнее режим конфигурации описан ниже. Итак, следует запомнить, что в любом из подрежимов можно воспользоваться интерактивной подсказкой, но ее результат будет разным. (Обратите внимание, настал момент задействовать симулятор NetSim Lite, — откройте любую лабораторную работу, используйте вопросительный знак и опробуйте несколько команд.)

Операционная система Cisco IOS сохраняет вводимые пользователем команды в так называемом буфере истории команд (стандартно в нем хранится десять запи-

сей). Интерфейс командной строки позволяет перемещаться вперед и назад по такому списку введенных команд и редактировать прежние команды перед их повторным введением. Необходимые для навигации по буферу истории команд клавиши и их комбинации помогут значительно быстрее перемещаться по командам на экзамене и в практической работе, поэтому их нужно запомнить. В табл. 7.4 перечислены клавиши и их комбинации для работы с буфером.

Таблица 7.4. Комбинации клавиш для вызова и редактирования команд из буфера

Комбинация клавиш	Выполняемое действие
<↑> или <Ctrl+P>	Отображает последнюю введенную команду, если нажать ее еще раз, предпоследнюю и так далее до тех пор, пока не закончится буфер (символ "P" означает "previous"; в переводе с англ. предыдущий)
<↓> или <Ctrl+N>	Если пользователь перешел на какую-либо из ранее введенных команд, эти клавиши перемещают вперед по буферу истории команд (символ "N" означает "next"; в переводе с англ. следующий)
<←> или <Ctrl+B>	Перемещает курсор влево (часто говорят "назад") по командной строке на один символ без удаления текста (символ "B" означает "back"; в переводе с англ. назад)
<→> или <Ctrl+F>	Перемещает курсор вправо ("вперед") по командной строке на один символ без удаления текста (символ "F" означает "forward"; в переводе с англ. вперед)
<Backspace>	Однократное нажатие удаляет последний введенный символ и перемещает курсор влево
<Ctrl+A>	Перемещает курсор в начало строки на самый первый символ команды
<Ctrl+E>	Перемещает курсор в конец командной строки
<Ctrl+R>	Повторно отображает командную строку. Эта комбинация клавиш полезна, когда выводимые устройством сообщения мешают вводить команду
<Ctrl+D>	Удаляет один символ
<Ctrl-Shift-6>	Прерывает текущую команду

Команды debug и show

Команда `show`, несомненно, — самая популярная в операционной системе Cisco IOS. У нее есть великое множество разнообразных параметров, позволяющих отследить состояние практически всех функций операционной системы, аппаратных модулей и т.п. Команда `show` отображает текущее состояние коммутатора, все, что устройство делает в ответ на любой вариант этой команды, — просто находит нужную информацию о состоянии чего-либо и отправляет ее в виде сообщений пользователю.

Команда `debug` играет роль, подобную команде `show`. Как и у команды `show`, у нее огромное количество параметров, но, в отличие от первой, принцип работы последней существенно отличается. Если команда `show` выдает некоторое текущее состояние какой-либо функции, другими словами, состояние устройства в какой-то определенный момент времени, то команда `debug` позволяет отслеживать функцию в процессе работы на протяжении некоего промежутка времени. Так, например, коммутатор может отправлять сообщения пользователю о каких-либо событиях, когда они происходят.

Команды show и debug можно сравнить с фотографией и с фильмом соответственно. Как и на фотографии, в выводе команды show показано что-то актуальное в определенный момент времени, другими словами, некоторая "застывшая картинка"; такая операция не требует много ресурсов. Команда debug показывает, как развивается процесс, что происходит и тому подобное (такой процесс требует много больше ресурсов). Вполне очевидно, что команда debug требует значительно больше времени центрального процессора, и в этом состоит ее основной недостаток, в то же время она показывает, что происходит в момент, когда происходит событие.

Сообщения от команд show и debug обрабатываются очень по-разному. Операционная система IOS посылает вывод команд show тому пользователю, который ее ввел, и никому другому. При вводе команды debug операционная система IOS создает регистрационные сообщения, зависящие от ее параметров. Любой зарегистрированный пользователь может просмотреть регистрационные сообщения при помощи команды terminal monitor из привилегированного режима.

Операционная система IOS считает команду show очень коротким событием, а команду debug — продолжительным. Если вывод какой-либо информации был включен командой debug с параметрами, то сообщения будут появляться до тех пор, пока пользователь не отключит их явно или не перезагрузит устройство. Перезагрузка (командой reload) устройства отключает все варианты отладки. Чтобы отключить вывод каких-либо определенных отладочных сообщений, следует использовать ту же команду debug с теми же параметрами и ключевым словом no в начале командной строки. Например, пользователь ввел команду **debug spanning-tree**; чтобы ее отключить, нужно ввести команду **no debug spanning-tree**. Можно также использовать команды no debug all и undebug all, чтобы отключить абсолютно все отладочные сообщения, которые были заданы ранее.

Следует помнить, что некоторые параметры команды debug создают огромное количество отладочных сообщений и, как следствие, система Cisco IOS не может их обработать или происходит аварийный отказ операционной системы. Просмотреть загрузку процессора устройства можно с помощью команды show process, это рекомендуется выполнить до запуска команд отладки. Чтобы быть уверенным в том, что устройство не "зависнет" или вдруг не перезагрузится от большой нагрузки, рекомендуется сначала ввести команду no debug all, а потом команду debug с параметрами. Если в результате выполнения второй команды устройство начнет работать неустойчиво, можно с помощью клавиши <↑> или комбинации <Ctrl+P> вызвать команду no debug all из буфера и применить ее. Если производительность устройства из-за отладки значительно упала, коммутатор может быть "слишком занят", чтобы отслеживать то, что вводится в командной строке. Прием, описанный в последнем абзаце, сэкономит время системному администратору и, несомненно, поможет избежать аварийного отказа операционной системы и незатребованной перезагрузки устройства.

Настройка программного обеспечения Cisco IOS

Чтобы сдать сертификационный экзамен и достичь успеха в профессиональной деятельности, нужно уметь настроить коммутатор компании Cisco. В этом разделе описаны основные последовательности настройки различных функций, в том числе что такое файл конфигурации и где он может быть сохранен. Несмотря на то что

в текущем разделе основное внимание уделяется принципам настройки устройств, а не каким-либо конкретным командам, все приведенные здесь команды нужно запомнить на память для сдачи экзамена и четко себе представлять процесс настройки устройства.

Режим конфигурации (configuration mode) устройства представляет собой специализированный режим интерфейса командной строки, в чем-то похожий на обычный или привилегированный режим. Привилегированный режим позволяет вводить "неопасные" для работы устройства команды, просматривать различные характеристики и информацию о состоянии устройства. В привилегированном режиме имеется много больше команд по сравнению с пользовательским, в том числе и команды, которые могут привести к неработоспособности устройства. Тем не менее ни в пользовательском, ни в привилегированном режиме нельзя изменить настройки коммутатора. Только в режиме конфигурации устройства можно ввести конфигурационные команды, т.е. команды, сообщающие устройству, что нужно предпринять и как нужно что-то сделать. На рис. 7.7 показана взаимосвязь режимов конфигурации, пользовательского и привилегированного.

Рис. 7.7. Режимы конфигурации, пользовательский и привилегированный

Вводимые в режиме конфигурации команды изменяют активный (т.е. текущий) файл конфигурации. Изменения попадают в конфигурацию сразу же после нажатия клавиши <Enter> в конце команды, поэтому следует быть осторожным и аккуратным при вводе команд!

Подрежимы и контексты конфигурации

В режиме конфигурации коммутатора или маршрутизатора есть много подрежимов. Контекстные команды настройки переводят пользовательский интерфейс из одного подрежима конфигурации, или контекста, в другой. Такие команды указывают маршрутизатору, с чем будут связаны вводимые после них команды. Что еще более важно, текущий подрежим указывает коммутатору, какие именно команды могут быть введены, поэтому, нажав клавишу со знаком вопроса (?) на клавиатуре, чтобы получить подсказку, можно увидеть команды, относящиеся только к текущему контексту интерфейса командной строки.

ВНИМАНИЕ!

Контекстные команды настройки (context-setting) — это не стандартный термин компании Cisco, а термин автора, здесь он используется только для удобства описания различных режимов конфигурации.

В качестве примера контекстной команды настройки устройства рассмотрим команду `interface`. Пользовательский интерфейс коммутатора переключается

в режим конфигурации интерфейса после ввода команды, например, **interface FastEthernet 0/1** в режиме глобальной конфигурации устройства. Если в таком подрежиме вызвать встроенную подсказку, как обычно, в ней будут присутствовать только команды, характерные для интерфейсов Ethernet. Команды в таком режиме иногда называют *подкомандами* (subcommand), или, в данном случае, подкомандами конфигурации интерфейса. Зачастую также специфические режимы конфигурации называют *подрежимами* (submode). Стоит попрактиковаться в настройке настоящих коммутаторов и маршрутизаторов, тогда переходы между режимами станут более понятными. Теперь рассмотрим небольшой пример (см. пример 7.2) настройки устройства, в котором показаны такие моменты:

- переход из привилегированного режима в режим глобальной конфигурации устройства с помощью команды `configure terminal`;
- использование команды `hostname Fred` режима глобальной конфигурации для настройки названия устройства;
- переход из режима глобальной конфигурации в режим конфигурации консольной линии с помощью команды `line console 0`;
- установка простого метода аутентификации по паролю с помощью команды `password hope`;
- переход из режима конфигурации консольного порта в режим конфигурации интерфейса с помощью команды `interface`;
- установка скорости в 100 Мбит/с для интерфейса Fa0/1 с помощью команды `speed 100`;
- переход из режима конфигурации интерфейса обратно в привилегированный режим с помощью команды `exit`.

Пример 7.2. Переходы между различными режимами конфигурации

```
Switch#configure terminal
Switch(config)#hostname Fred
Fred(config)#line console 0
Fred(config-line)#password hope
Fred(config-line)#interface FastEthernet 0/1
Fred(config-if)#speed 100
Fred(config-if)#exit
Fred(config)#
```

Текст в скобках в приглашении командной строки устройства идентифицирует текущий режим конфигурации, например, после введения соответствующей команды в третьей строке примера 7.2 видно, что режим поменялся. Обозначение "`(config)`" сообщает пользователю, что он находится в режиме глобальной конфигурации. После выполнения команды `line console 0` текст в скобках меняется на "`config-line`", что оповещает пользователя о переходе в режим конфигурации линии. В табл. 7.5 перечислены наиболее часто встречающиеся в работе варианты приглашения командной строки, названия соответствующих режимов и контекстные команды для перехода в такие режимы.

Ключевая тема

Таблица 7.5. Режимы конфигурации коммутатора

Приглашение командной строки	Название режима	Команды для перехода в режим
`hostname(config)#`	Глобальной конфигурации	`configure terminal`
`hostname(config-line)#`	Конфигурации линии	`line console 0` `line vty 0 15`
`hostname(config-if)#`	Конфигурации интерфейса	`interface` *тип номер*
`hostname(vlan)#`	VLAN	`vlan` *номер*

Имеет смысл отработать переходы между различными режимами конфигурации на практике, чтобы овладеть необходимыми навыками. Но эти навыки можно получить, просто выполняя лабораторные работы по теме в последующих главах книги. На рис. 7.8 представлена большая часть переходов между глобальным режимом конфигурации и четырьмя подрежимами конфигурации, перечисленными в табл. 7.5.

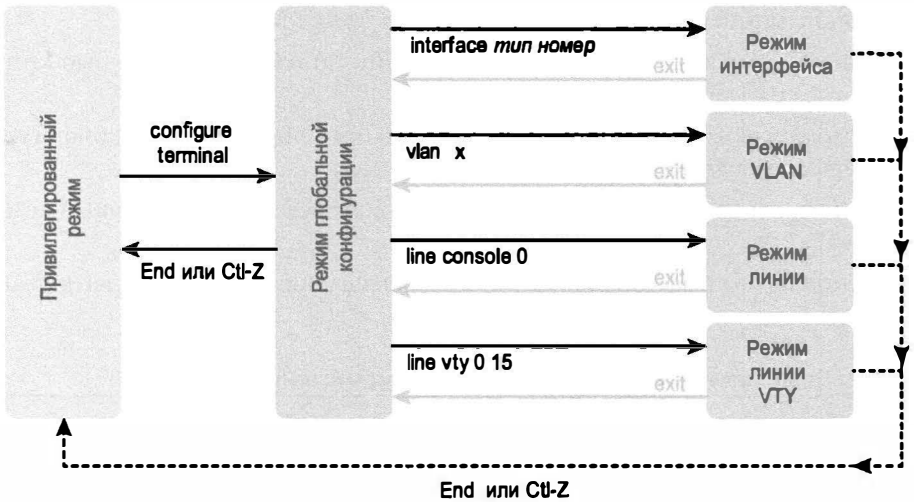


Рис. 7.8. Переходы между режимами конфигурации коммутатора

ВНИМАНИЕ!

Из одного подрежима конфигурации можно непосредственно перейти в другой без предварительной команды `exit`, возвращающей в глобальный режим конфигурации. Достаточно использовать команды в центре рисунка.

Каких-либо особых правил для глобальных команд или подкоманд не существует, тем не менее можно запомнить простое эмпирическое правило: если какой-то параметр может быть установлен на коммутаторе несколько раз, то соответствующая команда, вероятнее всего, вводится в каком-либо подрежиме. Если какая-то настройка может быть задана только один раз для всего коммутатора в целом, то, скорее всего, это команда глобального режима конфигурации. Например, команда `hostname` относится к глобальной настройке устройства, поскольку название

(hostname), или имя хоста, у коммутатора может быть только одно. Для сравнения: команда `duplex` является настройкой подрежима конфигурации интерфейса, следовательно, может быть введена несколько раз, и для различных интерфейсов могут быть указаны разные настройки.

Где хранятся файлы конфигурации?

Когда системный инженер настраивает коммутатор, он использует текущую конфигурацию. Текущую конфигурацию нужно где-то сохранить, чтобы при отсутствии питания можно было ее заново загрузить и использовать. В коммутаторах компании Cisco есть устройство оперативной памяти, используемое для хранения данных, необходимых операционной системе Cisco IOS. Все данные в оперативной памяти пропадают после перезагрузки или выключения питания устройства. Чтобы сохранить информацию, которая понадобится после выключения питания устройства, в коммутаторах компании Cisco используется несколько разновидностей постоянных запоминающих устройств, в которых нет никаких движущихся механических частей. За счет того, что механизмов в запоминающих устройствах нет (как в жестких дисках компьютеров, например), отказоустойчивость устройств заметно увеличивается и время наработки на отказ становится много больше.

Ниже перечислены четыре основных типа памяти, которые есть в коммутаторах компании Cisco, а также описано применение соответствующего хранилища.

- *Оперативная память* (Random Access Memory — RAM), иногда еще называемая *динамическим оперативным запоминающим устройством* (Dynamic Random-Access Memory — DRAM), используется в коммутаторе точно так же, как и в любом компьютере: для хранения текущей рабочей информации. *Текущая* (running), или *активная* (active), конфигурация устройства хранится в этой памяти.
- *Постоянное запоминающее устройство* (Read-Only Memory — ROM) хранит в себе *программу самозагрузки* (bootstrap или boothelper), которая срабатывает при включении питания устройства. Программа самозагрузки находит образ операционной системы Cisco IOS и управляет процессом его загрузки в оперативную память, после чего операционная система берет управление устройством на себя.
- *Флеш-память* (flash memory) представляет собой микросхему в коммутаторе, или съемный модуль памяти, в котором хранится полнофункциональный образ (или образы) операционной системы, и именно здесь процедура самозагрузки стандартно ищет систему. Во флеш-памяти могут также храниться другие файлы, например резервные копии файлов конфигурации.
- *Энергонезависимая память* (Nonvolatile RAM — NVRAM) используется для хранения *начальной*, или *стартовой* (startup), конфигурации устройства, использующейся при включении питания коммутатора и перезагрузке устройства.

На рис. 7.9 проиллюстрирована представленная выше информация, чтобы упростить запоминание материала.

Ключевая тема

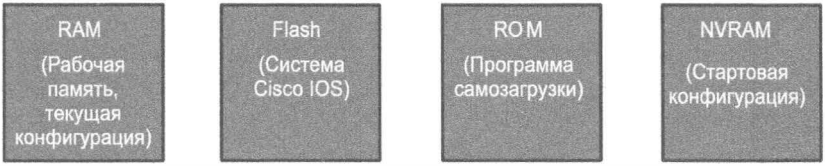


Рис. 7.9. Различные типы памяти коммутатора Cisco

Операционная система Cisco IOS сохраняет последовательность команд конфигурации в виде файла конфигурации. В действительности в коммутаторе есть два файла конфигурации: файл стартовой конфигурации, используемой при загрузке устройства, и файл текущей конфигурации, хранимый в оперативной памяти. В табл. 7.6 указаны оба файла конфигурации, их местоположение и предназначение.

Ключевая тема

Таблица 7.6. Названия и предназначение двух основных файлов конфигурации операционной системы Cisco IOS

Название файла конфигурации	Назначение	Место хранения
`startup-config`	Содержит стартовую конфигурацию, используемую каждый раз при загрузке и перезагрузке операционной системы Cisco IOS	NVRAM
`running-config`	Содержит текущие настройки устройства. Этот файл изменяется, когда кто-то вводит команды в режиме конфигурации устройства	RAM

По существу, когда системный инженер использует любой конфигурационный режим, он изменяет *файл текущей конфигурации* (`running-config`). Так, если вспомнить пример 7.2, то все вводимые в нем команды меняют только текущую конфигурацию устройств, и если после их ввода перезагрузить коммутатор (или если вдруг пропадет электропитание), вся новая конфигурация будет потеряна. Если же требуется, чтобы конфигурация сохранилась, то следует скопировать текущую конфигурацию в энергонезависимую память (NVRAM), т.е. перезаписать *файл стартовой конфигурации* (`startup-config`) устройства.

В примере 7.3 показано, как команды режима конфигурации устройства меняют только текущий файл конфигурации в оперативной памяти коммутатора. В этом примере продемонстрированы следующие этапы и концепции.

Этап 1 Проверить, что стартовая и текущая конфигурации совпадают, можно по параметру `hostname`

Этап 2 С помощью команды `hostname` инженер изменяет название устройства, но, согласно стандартному поведению коммутатора, только в файле текущей конфигурации (`running-config`)

Этап 3 На последнем этапе показан результат выполнения команд `show running-config` и `show startup-config` (показана настройка только названия устройства). По выводимой на экран информации можно определить, что два файла конфигурации отличаются

Пример 7.3. Команды конфигурации меняют только файл текущей конфигурации (running-config), но не стартовой (startup-config)

```
! Две команды, приведенные ниже, относятся к первому этапу
!
hannah# show running-config
! (строки конфигурации опущены)
hostname hannah
! (остальные строки также опущены)

hannah# show startup-config
! (строки конфигурации опущены)
hostname hannah
! (остальные строки также опущены)
! Ниже идут команды второго этапа.
! Обратите внимание, что приглашение командной строки меняется
! сразу же после ввода команды hostname.
!
hannah# configure terminal
hannah(config)#hostname jessie
jessie(config)#exit
! Две команды, приведенные ниже, относятся к третьему этапу
!
jessie# show running-config
! (строки конфигурации опущены)
hostname jessie
! (остальные строки также опущены)
! Обратите внимание, что в текущей конфигурации
! отображается измененное название устройства.
jessie# show startup-config
! (строки конфигурации опущены)
hostname hannah
! (остальные строки также опущены)
! Обратите внимание, что в стартовой конфигурации
! название устройства не поменялось.
```

ВНИМАНИЕ!

Для обозначения того, что в большинстве компьютерных операционных систем называется *перезапуском* (rebooting) или *перезагрузкой* (restarting), компания Cisco использует термин *перезагрузка* (reload). В любом случае это подразумевает повторную инициализацию программного обеспечения. Для перезагрузки коммутатора используется команда EXEC `reload`.

Копирование и удаление файлов конфигурации

Если необходимо сохранить новые команды конфигурации, введенные в режиме конфигурации (чтобы внесенные изменения остались после перезагрузки системы), такие, как команда `hostname jessie` в примере 7.3, необходимо использовать команду `copy running-config startup-config`. Эта команда переписывает файл стартовой конфигурации файлом текущей конфигурации.

Команда `copy` используется для настройки файлов на коммутаторе, обычно конфигураций, или образов, операционной системы Cisco IOS. Это наиболее простой способ передачи файлов в коммутатор и из него, причем перемещать файлы можно между оперативной и энергонезависимой памятью, сервером TFTP и т.п. Файлы могут быть скопированы между разными устройствами, как показано на рис. 7.10.

Общий формат копирования файлов в операционной системе Cisco IOS выглядит следующим образом:

```
copy {tftp | running-config | startup-config} {tftp | running-config |
startup-config}
```

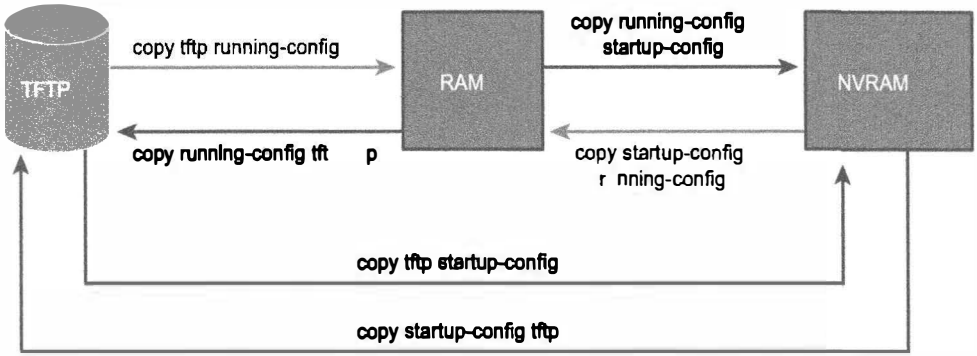


Рис. 7.10. Копирование файлов и результат этой операции

Первый параметр в фигурных скобках ({}) указывает, откуда копировать файл, второй — куда.

Команда `copy` всегда заменяет существующий файл на копируемый при перемещении файла в память NVRAM или на сервер TFTP. Другими словами, она удаляет файл, в который копируется источник, и заменяет его новым. При копировании файла из какого-либо источника в текущую конфигурацию (running-config) в оперативной памяти устройства файлы конфигурации не удаляются и не заменяются новыми, а объединяется, т.е. результирующая конфигурация будет комбинацией старого и нового файлов конфигурации. В действительности команда `copy` при копировании файла в оперативную память работает так, как если бы инженер вводил по очереди команды из нового файла конфигурации в интерфейсе командной строки.

"Ну, и какая разница, как работает команда?" — подумает читатель. Очень большая. Представим, что текущая конфигурация была изменена, но результат получился неудовлетворительным. Инженер решает вернуться к стартовой конфигурации и выполняет команду `copy startup-config running-config`, чтобы убрать неправильные настройки, но в результате два файла конфигурации, стартовой и текущей, все равно не будут совпадать. Единственным гарантированным методом возврата конфигурации в исходное состояние в этом случае будет команда `reload`, в результате выполнения которой коммутатор перезагрузится, все изменения в оперативной памяти будут потеряны, а при загрузке стартовая конфигурация будет скопирована в оперативную память, т.е. в файл текущей конфигурации.

Чтобы удалить содержимое энергонезависимой памяти (NVRAM), можно использовать три команды: `write erase` и `erase startup-config` (устаревшие варианты команды), `erase nvram:` (рекомендуемый современный вариант). Все указанные команды просто очищают память NVRAM и, следовательно, удаляют стартовый файл конфигурации. Если же сразу после удаления содержимого энергонезависимой памяти перезагрузить коммутатор, стартовой конфигурации не будет. Следует также запомнить, что у операционной системы компании Cisco нет команды для удаления текущей

конфигурации (`running-config`), поэтому для очистки текущей конфигурации необходимо удалить файл стартовой конфигурации (`startup-config`) и перезагрузить устройство (командой `reload`).

ВНИМАНИЕ!

Текущие конфигурации всех коммутаторов и маршрутизаторов в сети должны быть скопированы в безопасное место: на сетевой сервер, рабочую станцию сетевого инженера и т.п. Такой подход должен быть частью общей стратегии сетевой безопасности компании, поскольку позволит быстро заменить конфигурацию устройства в случае отказа или после хакерской атаки, в результате которой были изменены или повреждены настройки оборудования.

Диалог начальной конфигурации

Программное обеспечение IOS Cisco поддерживает два основных способа начальной базовой настройки коммутатора: режим конфигурации, который уже был рассмотрен в этой главе, и режим начальной конфигурации. *Режим начальной конфигурации* (setup mode) предлагает администратору коммутатора набор вопросов, ответы на которые позволяют задать параметры базовой конфигурации. После того как администратор ответит на вопросы, операционная система IOS создаст файл конфигурации, сохранит его в файле стартовой конфигурации, а также загрузит его как файл текущей конфигурации, чтобы сразу использовать новую конфигурацию.

Когда коммутатор или маршрутизатор Cisco инициализируется, но его файл стартовой конфигурации пуст, он спрашивает через консоль у пользователя, не хочет ли он настроить устройство. Схема процесса начальной конфигурации представлена на рис. 7.11.

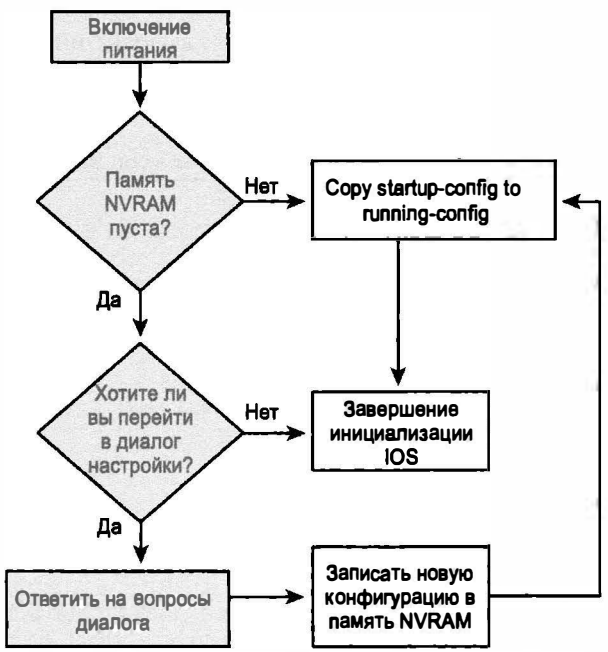


Рис. 7.11. Диалог начальной конфигурации устройства

Откровенно говоря, немногие сетевые инженеры используют режим начальной конфигурации, главным образом потому, что он позволяет настроить лишь небольшой процент параметров конфигурации современного коммутатора. Но он все еще применяется, ведь при перезагрузке коммутатора или маршрутизатора при отсутствии конфигурации операционная система IOS спросит, не хотите ли вы перейти в "диалог начальной конфигурации". Просто откажитесь, как показано в примере 7.4, и перейдите в режим конфигурации, позволяющий настроить устройство как нужно.

Пример 7.4. Пример диалога начальной конфигурации устройства (отказ)

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Switch>
```

Версия IOS и другие факты

В завершение первой главы, посвященной работе операционной системы Cisco IOS, рассмотрим команду коммутатора `show version`.

Во время загрузки операционной системы IOS коммутатор должен решить много задач. В оперативную память должно быть загружено само программное обеспечение IOS. Оно должно выяснить доступные аппаратные средства, например, типы всех интерфейсов LAN на коммутаторе. После этого программное обеспечение IOS собирает некую статистическую информацию о текущей работе коммутатора, например, время работы с момента последней загрузки и причина последней перезагрузки операционной системы.

Команда `show version` выводит информацию об этих и многих других фактах. Как следует из ее имени, команда `show version` выводит информацию об операционной системе IOS, включая ее версию. Однако, как подчеркнуто в примере 7.5, она выводит много и других интересных фактов.

Пример 7.5. Пример применения команды `show version` на коммутаторе Cisco

```
SW1# show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(1)SE3,
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 30-May-12 14:26 by prod_rel_team

ROM: Bootstrap program is C2960 boot loader
BOOTLDR: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(44)SE5, RELEASE
SOFTWARE (fc1)

SW1 uptime is 2 days, 22 hours, 2 minutes
System returned to ROM by power-on
System image file is "flash:c2960-lanbasek9-mz.150-1.SE3.bin"


This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and use…
! Строки опущены для краткости
```

```
cisco WS-C2960-24TT-L (PowerPC405) processor (revision P0) with 65536K
bytes of memory.
Processor board ID FCQ1621X6QC
Last reset from power-on
1 Virtual Ethernet interface
24 FastEthernet interfaces
2 Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.


64K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address       : 18:33:9D:7B:13:80
Motherboard assembly number     : 73-11473-11
Power supply part number        : 341-0097-03
Motherboard serial number       : FCQ162103ZL
Power supply serial number      : ALD1619B37W
Model revision number           : P0
Motherboard revision number     : A0
Model number                    : WS-C2960-24TT-L
System serial number            : FCQ1621X6QC
Top Assembly Part Number        : 800-29859-06
Top Assembly Revision Number    : C0
Version ID                      : V10
CLEI Code Number                : COMCX00ARB
Hardware Board Revision Number  : 0x01


Switch Ports Model              SW Version      SW Image
------ ----- -----              ----------      ----------
*    1 26    WS-C2960-24TT-L    15.0(1)SE3      C2960-LANBASEK9-M


Configuration register is 0xF
```

В выводе команды **show version** выделена следующая информация:

- версия операционной системы IOS;
- время работы после последней загрузки IOS;
- причина последней перезагрузки IOS;
- количество интерфейсов Fast Ethernet (24);
- количество интерфейсов Gigabit Ethernet (2);
- номер модели коммутатора.

Обзор

Резюме

- В маршрутизаторах и большинстве моделей коммутаторов Catalyst компании Cisco используется концепция *интерфейса командной строки*. CLI представляет собой текстовый интерфейс, в котором пользователь, обычно сетевой инженер, вводит некоторые команды в виде текста.
- Как и любые другие аппаратные средства, коммутаторы Cisco нуждаются в программном обеспечении — операционной системе. Компания Cisco использует операционную систему IOS.
- Получить доступ к интерфейсу командной строки можно с помощью трех популярных методов: через консольное подключение, через протокол Telnet и различные варианты программы протокола SSH. Консольное подключение — это специализированный физический порт устройства для доступа к интерфейсу командной строки и настройки.
- Физическое консольное соединение, и прежнее, и новое, использует три основных компонента: физический консольный порт на коммутаторе, физический последовательный порт на компьютере и кабель, соединяющий консольный и последовательный порты.
- После физического подключения компьютера к консольному порту на компьютере следует установить и настроить пакет программ эмулятора терминала. Эмулятор рассматривает все данные как текст. Он принимает текст, введенный пользователем, и посылает его по консольному соединению на коммутатор. Точно так же все биты, поступающие на компьютер по консольному соединению, отображаются как текст для удобства чтения пользователем.
- Эмулятор терминала нужно настроить так, чтобы он использовал последовательный порт компьютера, а его настройки должны совпадать со стандартными настройками коммутатора. Последние три настройки по первым буквам параметров для простого запоминания записывают как "8N1":
  - скорость 9600 бит/с;
  - без аппаратного контроля потока;
  - 8 бит данных;
  - без контроля четности;
  - 1 стоповый бит.
- Коммутатор Cisco со стандартными настройками остается относительно защищенным, будучи закрытым в шкафу кабельного узла, поскольку допускает управление только через консоль. Но когда разрешен доступ по технологиям Telnet и (или) SSH, необходимо установить пароль, чтобы только уполномоченные лица имели доступ к его интерфейсу CLI.
- Пользовательский режим, или режим EXEC обычного пользователя. В этом режиме пользователь преимущественно может просматривать различную

информацию, но не настроить или "сломать" что-то. Приставка "EXEC" подчеркивает, что когда пользователь вводит какую-либо команду, коммутатор ее выполняет и выводит некоторое сообщение о результатах работы.

- Операционная система Cisco IOS поддерживает режим EXEC с большими возможностями — привилегированный режим, он же режим enable, или привилегированный режим EXEC. Привилегированный режим получил свое название от команды `enable`, переводящей пользователя из пользовательского режима в привилегированный.
- Вид подсказки зависит от текущего режима интерфейса командной строки. Например, если нажать <?> в режиме обычного (непривилегированного) пользователя, будут показаны команды, которые разрешено выполнять из этого режима, а команды из привилегированного режима отображаться не будут. Иначе будет также выглядеть интерактивная подсказка при использовании ее в режиме конфигурации — как минимум, она будет показывать совсем другие команды.
- Операционная система Cisco IOS сохраняет вводимые пользователем команды в так называемом буфере истории команд (стандартно в нем хранится десять записей). Интерфейс командной строки позволяет перемещаться вперед и назад по такому списку введенных команд и редактировать прежние команды перед их повторным введением.
- Команда `show`, несомненно, — самая популярная в операционной системе Cisco IOS. У нее есть великое множество разнообразных параметров, позволяющих отследить состояние практически всех функций операционной системы, аппаратных модулей и т.п.
- Команда `debug` играет роль, подобную команде `show`. Как и у команды `show`, у нее огромное количество параметров, но, в отличие от первой, принцип работы последней существенно отличается. Если команда `show` выдает некоторое текущее состояние какой-либо функции, другими словами, состояние устройства в какой-то определенный момент времени, то команда `debug` позволяет отслеживать функцию в процессе работы на протяжении некоего промежутка времени.
- *Режим конфигурации* устройства представляет собой специализированный режим интерфейса командной строки, в чем-то похожий на обычный или привилегированный режим. Привилегированный режим позволяет вводить "неопасные" для работы устройства команды, просматривать различные характеристики и информацию о состоянии устройства. В привилегированном режиме имеется много больше команд по сравнению с пользовательским, в том числе и команды, которые могут привести к неработоспособности устройства. Тем не менее ни в пользовательском, ни в привилегированном режиме нельзя изменить настройки коммутатора.
- В режиме конфигурации коммутатора или маршрутизатора есть много подрежимов. Контекстные команды настройки переводят пользовательский интерфейс из одного подрежима конфигурации, или контекста, в другой.

- Команда `interface` — наиболее популярный пример контекстных команд. Режим конфигурации интерфейса после ввода команды, например **`interface FastEthernet 0/1`**, переключается в режиме глобальной конфигурации устройства.
- Файл `startup-config` содержит стартовую конфигурацию, используемую каждый раз при загрузке и перезагрузке операционной системы Cisco IOS. Хранится в энергонезависимой памяти (NVRAM).
- Файл `running-config` содержит текущие настройки устройства. Этот файл изменяется, когда кто-то вводит команды в режиме конфигурации устройства. Хранится в памяти RAM.
- Если необходимо сохранить новые команды конфигурации, введенные в режиме конфигурации (чтобы внесенные изменения остались после перезагрузки системы), необходимо использовать команду `copy running-config startup-config`.
- Режим начальной конфигурации предлагает администратору коммутатора набор вопросов, ответы на которые позволяют задать параметры базовой конфигурации. После того как администратор ответит на вопросы, операционная система IOS создаст файл конфигурации, сохранит его в файле стартовой конфигурации, а также загрузит его как файл текущей конфигурации, чтобы сразу использовать новую конфигурацию.
- Команда `show version` сообщает следующую информацию:
  - версия операционной системы IOS;
  - время работы после последней загрузки IOS;
  - причина последней перезагрузки IOS;
  - количество интерфейсов Fast Ethernet (24);
  - количество интерфейсов Gigabit Ethernet (2);
  - номер модели коммутатора.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Из какого режима можно выполнить команду `show mac-address-table`? (Выберите два ответа.)

   А) Из обычного режима (User).

   Б) Из привилегированного режима (Enable).

   В) Из режима глобальной конфигурации (Global configuration).

   Г) Из режима начальной настройки устройства (Setup).

   Д) Из режима конфигурации интерфейса (Interface configuration).

2. В каком из указанных ниже режимов командной строки можно выполнить команду перезагрузки устройства?

А) Из обычного режима (user).

Б) Из привилегированного режима (enable).

В) Из режима глобальной конфигурации (global configuration).

Г) Из режима конфигурации интерфейса (interface configuration).

3. Коммутаторы компании Cisco поддерживают доступ через сеансы Telnet и SSH. Чем отличаются эти протоколы?

А) В протоколе SSH шифруется пароль, используемый при аутентификации, остальной трафик — нет; в протоколе Telnet ничего не шифруется.

Б) В протоколе SSH шифруется весь обмен данными, в том числе и пароли для аутентификации; в протоколе Telnet ничего не шифруется.

В) Протокол Telnet используется операционными системами от компании Microsoft; протокол SSH можно использовать только в операционных системах, подобных Unix, в частности в Linux.

Г) В протоколе Telnet шифруется только пароль; в протоколе SSH шифруется весь обмен данными.

4. В какой памяти хранится используемая коммутатором конфигурация в процессе его работы?

А) RAM.

Б) ROM.

В) Flash.

Г) NVRAM.

Д) Bubble.

5. С помощью какой команды можно скопировать конфигурацию из оперативной памяти RAM в энергонезависимую память NVRAM?

А) `copy running-config tftp`.

Б) `copy tftp running-config`.

В) `copy running-config start-up-config`.

Г) `copy start-up-config running-config`.

Д) `copy startup-config running-config`.

Е) `copy running-config startup-config`.

6. Пользователь находится в режиме конфигурации консольной линии в интерфейсе командной строки. Какая последовательность действий переведет его в привилегированный режим? (Выберите два ответа.)

А) Необходимо однократно использовать команду `exit`.

Б) Следует указать команду `exit` два раза подряд в одной командной строке.

В) Нужно нажать комбинацию клавиш <Ctrl+z>.

Г) Следует использовать команду `quit`.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 7.7.

Таблица 7.7. Ключевые темы главы 7

Элемент	Описание	Страница
Рис. 7.4	Возможности кабельного подключения к консоли	223
Список	Стандартные настройки консольного порта коммутатора компании Cisco	224
Табл. 7.5	Режимы конфигурации коммутатора	234
Рис. 7.9	Различные типы памяти коммутатора Cisco	236
Табл. 7.6	Названия и предназначение двух основных файлов конфигурации операционной системы Cisco IOS	236

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

интерфейс командной строки (Command-Line Interface — CLI), Telnet, защищенное удаленное соединение (Secure Shell — SSH), привилегированный режим (enable mode), пользовательский режим (user mode), режим конфигурации (configuration mode), файл `tartup-config`, файл `running-config`

Таблицы команд

В табл. 7.8 приведены команды конфигурации этой главы с описанием, а в табл. 7.9 — список пользовательских команд.

Таблица 7.8. Команды конфигурации главы 7

Команда	Режим и назначение
`line console 0`	Переключает контекст командной строки в режим конфигурации консольной линии
`line vty` *1-я_линия_vty последняя_линия_vty*	Переключает контекст в режим конфигурации линий vty для указанного в команде диапазона
`login`	Режим конфигурации консольной линии или линии vty. Указывает операционной системе IOS, что нужно выводить приглашение для ввода пароля
`password` *пароль*	Режим конфигурации консольной линии или линии vty. Задает пароль, который будет запрашиваться при подключении к устройству
`interface` *тип номер_порта*	Переключает контекст командной строки в режим конфигурации интерфейса. В параметре типа интерфейса обычно указывают значение FastEthernet или GigabitEthernet (для коммутаторов). Возможные номера портов зависят от модели коммутатора и выглядят, например, так: 0/1, 0/2 и т.п.

Окончание табл. 7.8

Команда	Режим и назначение
`hostname` *имя*	Режим глобальной конфигурации. Задает имя узла для коммутатора, которое также используется в приглашении командной строки
`exit`	Переводит интерфейс командной строки на один подрежим выше в режиме конфигурации чего-либо
`end`	Переводит интерфейс командной строки из любого подрежима конфигурации в привилегированный режим
<Ctrl+Z>	Эта комбинация клавиш делает то же, что и команда `end`

Таблица 7.9. Пользовательские команды главы 7

`no debug all` `undebug all`	Команды привилегированного уровня, отключающие все активные отладки в устройстве
`terminal monitor`	Указывает операционной системе Cisco IOS, что копии всех системных сообщений, в том числе и отладочных, нужно пересылать в сеанс Telnet или SSH пользователя
`reload`	Привилегированный режим. Команда вызывает перезагрузку коммутатора или маршрутизатора
`copy` *откуда куда*	Привилегированный режим. Команда копирует файл из одного местоположения в другое. В качестве источника и получателя файла могут использоваться текущая и стартовая конфигурация, сервер TFTP и RCP, флеш-память
`copy running-config` `startup-config`	Привилегированный режим. Команда сохраняет активную текущую конфигурацию в стартовой, перезаписывая соответствующий файл. Стартовая конфигурация используется при загрузке устройства
`copy startup-config` `running-config`	Привилегированный режим. Команда объединяет файл стартовой конфигурации и текущей (в оперативной памяти)
`show running-config`	Показывает файл текущей конфигурации устройства
`write erase` `erase startup-config`	Все указанные команды в привилегированном режиме позволяют полностью удалить стартовую конфигурацию
`quit`	Команда непривилегированного режима. Разрывает сеанс с интерфейсом командной строки
`show system:running-config`	Аналог команды `show running-config`
`show startup-config`	Показывает файл стартовой конфигурации устройства
`show nvram:startup-config` `show nvram:`	Аналоги команды `show startup-config`
`enable`	Переводит из режима обычного пользователя в привилегированный и запрашивает пароль, если он задан
`disable`	Переводит пользователя из привилегированного режима в обычный
`configure terminal`	Команда привилегированного режима. Переводит в режим конфигурации устройства

Ответы на контрольные вопросы:
1 А и Б. **2** Б. **3** Б. **4** А. **5** Е. **6** Б и В.

Настройка коммутаторов Ethernet

Коммутаторы Cisco локальных сетей способны выполнять свою основную задачу (перенаправление фреймов Ethernet) без всякой настройки. Вполне можно купить коммутатор Cisco, подключить к нему соответствующие кабели от различных устройств, подключить питание, и коммутатор заработает. Но в большинстве сетей сетевому инженеру нужно настроить и использовать разные средства коммутатора.

Различные средства коммутатора рассматриваются в этой главе. Сначала обсуждается набор средств администрирования, одинаково применимых как на маршрутизаторах, так и на коммутаторах; в этой главе они собраны вместе, чтобы впоследствии, при работе с маршрутизаторами, их описание было проще найти. Затем будет продемонстрирована настройка некоторых специфических средств коммутатора, большинство из которых влияет на перенаправление фреймов коммутатором.

В этой главе рассматриваются следующие экзаменационные темы

Технологии маршрутизации IP
Настройка интерфейсов SVI.
Защита сетевых устройств
Настройка и проверка средств защиты сетевых устройств.
  Защита устройства паролем.
  Привилегированный режим или защита.
  SSH.
  VTY.
  Служебный пароль.
  Описание основных методов аутентификации.
Настройка и проверка средств защиты порта коммутатора.
  Автоматическое обнаружение MAC-адресов.
  Ограничение MAC-адресов.
  Статические и динамические.
  Реакция при нарушении защиты.
    Отключение из-за ошибки.
    Отключение.
    Ограничение.
  Отключение неиспользуемых портов.
  Восстановление после ошибки.
  Присвоение неиспользуемых портов неиспользуемым VLAN.

Основные темы

Настройка функций, общих для коммутаторов и маршрутизаторов

В первом из двух разделов этой главы подробно описана настройка функций, одинаково применимых для коммутаторов и маршрутизаторов. В частности, будут подробно рассмотрены настройки защиты доступа к интерфейсу командной строки, а также настройка различных параметров консольного подключения к устройству. И хотя этот раздел посвящен коммутаторам, а не маршрутизаторам, рассматриваемые команды применимы к ним обоим.

Защищенный доступ к командной строке коммутатора

Первый этап защиты коммутатора — это защита доступа к интерфейсу CLI. Она подразумевает защиту доступа к привилегированному режиму, поскольку именно из него злоумышленник может перезагрузить коммутатор или изменить его конфигурацию. В то же время защита пользовательского режима также важна, поскольку злоумышленник может просмотреть состояние коммутатора и, узнав информацию о сети, найти новые способы атаки на нее.

Рассмотрим пример пользователя, обращающегося к коммутатору по консоли. Стандартная конфигурация консоли позволяют консольному пользователю перейти из пользовательского режима в привилегированный, не вводя пароль. Это имеет смысл, поскольку при использовании консоли сетевой инженер, как правило, находится рядом с коммутатором. Если коммутатор доступен физически, то защита консоли паролем преодолевается процедурой сброса и восстановления пароля минут за пять, так или иначе, консоль коммутатора станет доступна. Поэтому изначально консольный доступ является открытым. Однако большинство сетевых инженеров устанавливают защиту и на вход в систему консоли.

ВНИМАНИЕ!

Чтобы найти процедуры сброса и восстановления паролей, обратитесь к веб-сайту `Cisco.com` и выполните поиск по фразе “password recovery” (восстановление пароля). Первый же результат на странице поиска, вполне вероятно, будет относиться к описанию процедуры восстановления паролей практически для всех устройств компании Cisco.

С другой стороны, стандартные параметры конфигурации коммутатора не разрешают сеансы vty (Telnet или SSH) ни в пользовательском, ни в привилегированном режиме. Чтобы позволить этим пользователям перейти в пользовательский режим, коммутатору требуется рабочая конфигурация IP-адреса, а также защита входа в систему на линиях vty. Для предоставления доступа к привилегированному режиму на коммутаторе должна быть также настроена защита привилегированного режима.

В данном разделе рассматриваются подробности конфигурации, связанные с доступом пользователя к привилегированному режиму коммутатора или маршрутизатора. Настройка IP-адреса коммутатора рассматривается далее в этой главе. В частности, в настоящем разделе затрагиваются следующие темы:

- простой пароль для пользовательского режима консоли и сеансов Telnet;
- защищенное дистанционное соединение (SSH);
- шифрование паролей;
- пароли привилегированного режима.

Зашита доступа простым паролем

Для пользователей Telnet и консоли коммутаторы Cisco способны защитить пользовательский режим простым паролем (без имени пользователя). Пользователи консоли должны ввести *пароль консоли* (console password), заданный в режиме конфигурации линии консоли. Пользователи Telnet должны вводить *пароль Telnet* (Telnet password), называемый также *паролем vty* (vty password), поскольку его конфигурация находится в режиме конфигурации линии vty.

Коммутаторы Cisco защищают привилегированный режим при помощи *привилегированного пароля* (enable password). Пользователь в пользовательском режиме вводит команду `enable`, запрашивающую привилегированный пароль; если пользователь вводит правильный пароль, операционная система IOS переводит пользователя в привилегированный режим. На рис. 8.1 приведены названия этих паролей и связанных с ними режимов конфигурации.

Рис. 8.1. Концепции простой защиты паролем

Настройка этих паролей не требует большого труда. Настройка паролей консоли и vty использует те же две подкоманды консоли и vty соответственно, в режиме конфигурации линии. Команда `login` указывает операционной системе IOS использовать простой пароль, а команда `password` *`пароль_значение`* задает пароль. Операционная система IOS защищает привилегированный режим, используя привилегированный пароль, заданный глобальной командой `enable secret` *`пароль_значение`*.

ВНИМАНИЕ!

Ниже, в разделе "Предоставление IP-адреса для дистанционного доступа", подробно описаны два варианта настройки паролей привилегированного режима с помощью команд `enable`, `enable secret` и `enable password`, а также объясняется, почему первый вариант предпочтительней.

В примере 8.1 показан процесс настройки пароля для консольного (console) подключения, пароля каналов vty (сеансов Telnet), шифрованного привилегированного пароля (enable) и настройки имени хоста (hostname) для коммутатора. В примере показан процесс целиком, в том числе и внешний вид приглашения командной строки, подробно описанный в главе 7.

Пример 8.1. Настройка простых паролей и имен хостов

Ключевая тема

```
Switch> enable
Switch# configure terminal
Switch(config)# enable secret cisco
Switch(config)# hostname Emma
Emma(config)# line console 0
Emma(config-line)# password faith
Emma(config-line)# login
Emma(config-line)# exit
Emma(config)# line vty 0 15
Emma(config-line)# password love
Emma(config-line)# login
Emma(config-line)# exit
Emma(config)# exit
Emma#
```

Поскольку самостоятельно вы, вероятно, настроили еще не много конфигураций, ниже мы рассмотрим пример 8.1 построчно, используя его для демонстрации работы интерфейса CLI. Сначала сосредоточимся на первых четырех строках, приглашения к вводу которых начинаются со `Switch`. В соответствии с соглашением об оформлении книги весь текстовый вывод отображается обычным, не полужирным шрифтом, а введенный пользователем — полужирным. Например, первая строка демонстрирует приглашение к вводу команд коммутатора `Switch>` (стандартное приглашение), а пользователь ввел команду **`enable`**. Символ > в конце приглашения указывает на то, что пользователь находится в пользовательском режиме.

Эти первые несколько строк демонстрируют переход пользователя из пользовательского режима в привилегированный (команда `enable`). Затем пользователь переходит в глобальный режим конфигурации (команда `configure terminal`). Оказавшись в глобальном режиме конфигурации, пользователь вводит две команды (`enable secret` и `hostname`), распространяющиеся на весь коммутатор.

Первая строка вывода примера 8.1, начинающаяся с `Emma`, демонстрирует начало настройки пароля консоли. Сначала, используя команду `line console 0`, пользователь должен войти в режим конфигурации канала консоли. (Коммутатор имеет только один канал консоли, и он всегда имеет номер 0.) Затем команда `password` задает простой текстовый пароль (`faith`), а команда `login` указывает коммутатору запрашивать его, как определено командой пароля.

Следующие строки примера повторяют те же действия, но для всех 16 каналов vty (линии vty от 0 до 15). 16 каналов vty означает, что коммутатор может принять 16 параллельных подключений Telnet к коммутатору. Что касается пароля, то конфигурация использует для линий vty другой пароль — "`love`". И наконец, команда `end` возвращает пользователя в привилегированный режим.

Перейдем от конфигурации к тому, что будут видеть пользователи благодаря настройке в примере 8.1. Теперь у пользователя консоли будет запрашиваться пароль (без имени пользователя), и он должен ввести `hope`. Точно так же у пользователей Telnet будет запрашиваться пароль (тоже без имени пользователя), и он должен будет ввести `love`. Для перехода в привилегированный режим пользователи консоли

и Telnet должны использовать команду `enable` с паролем "`cisco`". Пользователи SSH пока не смогут войти на этот коммутатор, поскольку для поддержки протокола SSH необходимо больше действий.

Пример 8.2 демонстрирует полученную в результате конфигурацию на коммутаторе `Emma`. Серым выделены строки новой конфигурации. Обратите внимание, что некоторые строки вывода коммутатора были опущены, чтобы уменьшить объем строк, не имеющих отношения к теме.

Пример 8.2. Полученный файл `running-config` на коммутаторе Emma

```
Emma# show running-config
!
Building configuration...

Current configuration : 1333 bytes
!
version 12.2
!
hostname Emma
!
enable secret 5 $1$YXRN$11zOe1Lb0Lv/nHyTquobd.
!
spanning-tree mode pvst
spanning-tree extend system-id
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
! Некоторые строки конфигурации были опущены, в частности,
! настройки интерфейсов FastEthernet с 0/3 по 0/23.
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
no ip address
no ip route-cache
!
!
line con 0
  password faith
  login
!
line vty 0 4
  password love
  login
!
line vty 5 15
  password love
  login
```

ВНИМАНИЕ!

В последних шести строках примера 8.2 вывод команды `show running-config` отделяет первые пять каналов vty (0–4) от остальных (5–15) по исторически сложившимся причинам.

Зашита доступа по локальному имени пользователя и паролю

Метод доступа по простому текстовому паролю (без имен пользователя) вполне работоспособен, но он требует, чтобы все знали те же пароли. Например, для доступа к коммутатору по Telnet все должны знать тот же пароль vty.

Коммутаторы Cisco поддерживают и другие методы аутентификации, подразумевающие использование имени пользователя и пароля, чтобы у каждого пользователя был собственный пароль, а не общий. Имя пользователя и пароль можно настроить локально на коммутаторе, а можно положиться на внешний сервер — сервер *аутентификации, авторизации и учета* (Authentication, Authorization, And Accounting — AAA. (Это может быть тот же сервер, который используется для регистрации других серверов в сети.) В этой книге рассматривается конфигурация с именами пользователя и паролями, заданными локально.

Переход от доступа только по паролю к локально заданным именам пользователя и паролям требует лишь немногих изменений в конфигурации. Для этого достаточно одной или нескольких глобальных команд конфигурации `username` *имя* `password` *пароль*. Затем нужно уведомить каналы консоли и vty об использовании заданных имен пользователя и пароля (подкоманда линии `login local`). На рис. 8.2 приведена концепция перехода и пример конфигурации локальных имен пользователя и паролей для пользователей Telnet.

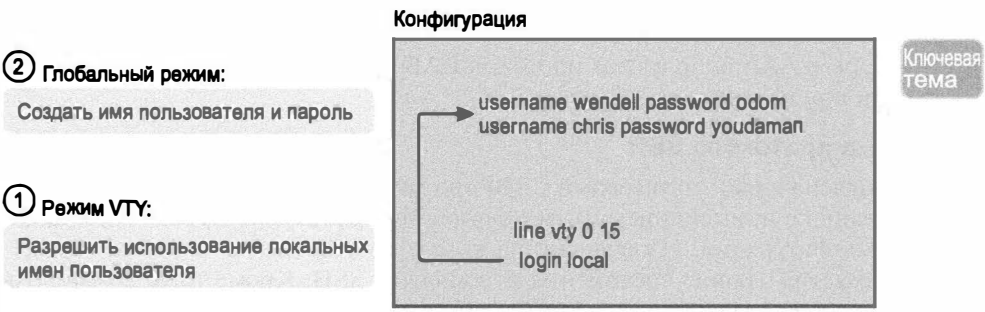


Рис. 8.2. Настройка на коммутаторе аутентификации по локальному имени пользователя и паролю

Согласно конфигурации на рис. 8.2, при попытке пользователя Telnet подключиться к коммутатору у него сначала будет запрошено имя, а затем пароль. Если соответствующей пары имени пользователя и пароля нет в списке коммутатора, то попытка доступа отклоняется.

ВНИМАНИЕ!

Команды конфигурации на рис. 8.2 приведены в том же порядке, что и для команды IOS `show running-config`.

Зашита доступа при помощи внешних серверов аутентификации

Локальный список имен пользователя и паролей на коммутаторе или маршрутизаторе хорошо работает в малых сетях. Однако использование локальных пар имени пользователя и пароля означает настройку на каждом коммутаторе и маршрутизаторе конфигурации для всех пользователей, которым возможно понадобится доступ к устройству. Если впоследствии понадобится внести какие-либо изменения, придется менять конфигурацию каждого устройства.

Коммутаторы и маршрутизаторы Cisco поддерживают еще один способ проверки правильности имен пользователя и паролей — внешний сервер AAA. При этом подходе аутентификации коммутатор (или маршрутизатор) просто посылает на сервер AAA сообщение с запросом, допустимы ли данное имя пользователя и пароль, а сервер AAA отвечает. Пример приведен на рис. 8.3: пользователь вводит свое имя и пароль, коммутатор запрашивает сервер AAA, и он отвечает коммутатору, сообщая, что имя пользователя и пароль допустимы.

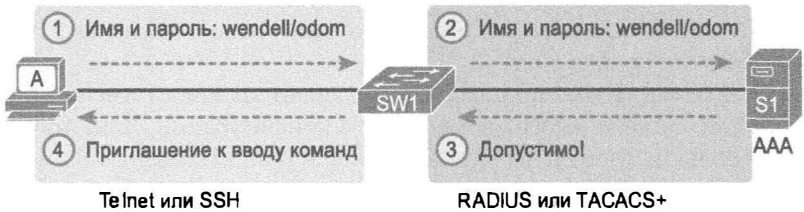


Рис. 8.3. Процесс простой аутентификации при внешнем сервере AAA

Обратите внимание на то, что потоки информации на рисунке передаются по нескольким разным протоколам. Слева соединение между пользователем и коммутатором или маршрутизатором использует протокол Telnet или SSH. Справа коммутатор и сервер AAA используют протокол RADIUS или TACACS+, оба они шифруют пароли при передаче по сети.

Настройка протокола SSH

Для поддержки коммутаторами Cisco протокола SSH требуется не только базовая конфигурация с применением имен пользователей и паролей Telnet, но и дополнительная конфигурация. Изначально на коммутаторе уже выполняется сервер SSH, он принимает входящие соединения от клиентов SSH. Кроме того, коммутатор нуждается в *криптографическом ключе* (cryptography key), используемом для шифрования данных. Ниже приведены этапы настройки поддержки протокола SSH на коммутаторе Cisco, использующем локальные имена пользователей и пароли. Специфические для поддержки протокола SSH команды применяют на этапе 3.

Ключевая тема

Этапы настройки протокола SSH на коммутаторе

Этап 1 Настройте линии vty на использование имен пользователя локально или на сервере AAA (используя команду `login local`)

Этап 2 При использовании локальных имен пользователя добавьте одну или несколько глобальных команд конфигурации `username`, чтобы задать пары имен пользователей и паролей

Этап 3 Настройте коммутатор на создание соответствующих пар открытых и закрытых ключей, используемых при шифровании. Для этого используются две команды:

A. В качестве предпосылки для следующей команды задайте имя домена DNS при помощи глобальной команды конфигурации `ip domain-name` *имя*.

B. Создайте ключи шифрования, используя глобальную команду конфигурации `crypto key generate rsa`

Этап 4 Для повышения защиты разрешите использование версии 2 протокола SSH, используя глобальную команду `ip ssh version 2` (Необязательно.)

Рис. 8.4 демонстрирует три этапа с примерами необходимых команд конфигурации. Обратите внимание, что код на рисунке добавляет к конфигурации, представленной на рис. 8.2, еще две команды, обеспечивающие поддержку SSH.

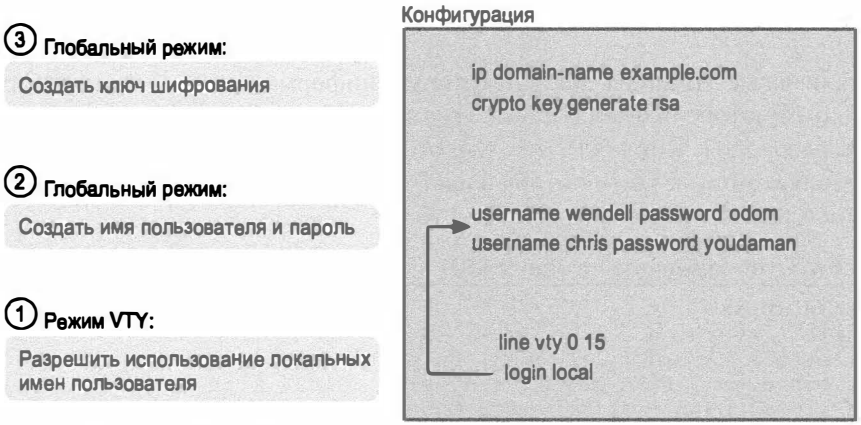


Рис. 8.4. Настройка на коммутаторе поддержки протокола SSH

ВНИМАНИЕ!

Этапы на рисунке приведены снизу вверх, поскольку вывод команды `show running-config` отобразит команды конфигурации именно в этом порядке.

Пошаговый просмотр настроек в режиме конфигурации может быть особенно полезен в случае протокола SSH. Обратите, в частности, внимание на то, что во время создания ключа команда `crypto key` фактически запрашивает у пользователя дополнительную информацию и выдает некие сообщения. Пример 8.3 демонстрирует команды конфигурации рис. 8.4 с ключом шифрования на последнем этапе.

Пример 8.3. Процесс настройки протокола SSH

```
Emma# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Emma(config)# line vty 0 15
! Ниже вводится команда этапа 1
Emma(config-line)# login local
Emma(config-line)# exit
!
! Ниже вводится команда этапа 2
Emma(config)# username wendell password odom
```

```
Emma(config)# username chris password youdaman
!
! Ниже вводится команда этапа 3
Emma(config)# ip domain-name example.com
Emma(config)# crypto key generate rsa
The name for the keys will be: Emma.example.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 4 seconds)

Emma(config)# ip ssh version 2
Emma(config)# ^Z
Emma#
```

Две ключевые команды дают некоторую информацию о состоянии протокола SSH на коммутаторе. Команда `show ip ssh` отображает информацию о состоянии самого сервера SSH. Команда `show ssh` отображает информацию о каждом клиенте SSH, подключенном к коммутатору в настоящее время. В примере 8.4 они приведены для пользователя Wendell.

Пример 8.4. Отображение состояния SSH

```
Emma# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQC+/mp2iaeaGwjqkIgLNH+lN/04LTc2u6qHVHHv3hoq
/DDBd9vABNnJGsq8z0Hm9HcrSudC20N/cCuEb4x5+T9rvNkUeAqwEEoJALpdiWVOpBliomhPy
svJi+m4
wI16AH31KI+GFCZv1AIjZSYHQEbvdCEqsYezAeKnPhvzTrUqaQ==


Emma# show ssh
Connection Version Mode Encryption Hmac      State           Username
0          2.0     IN   aes128-cbc hmac-sha1 Session started wendell
0          2.0     OUT  aes128-cbc hmac-sha1 Session started wendell
%No SSHv1 server connections running.
```

В этом примере использована версия 2 протокола SSH, а не версия 1. У нового протокола SSH лучше базовые алгоритмы защиты по сравнению с протоколом SSH v1, а также добавлены некоторые другие преимущества, например поддержка сообщений.

И наконец, коммутатор поддерживает на линиях vty доступ по протоколам Telnet и SSH, но для повышения безопасности можно отключить один или оба из них. Например, руководство компании могло бы потребовать запретить доступ по протоколу Telnet из-за риска безопасности. Поэтому необходимо отключить поддержку протокола Telnet на коммутаторе. Коммутатор контролирует поддержку протоколов Telnet и SSH на линиях vty, используя подкоманду vty `transport input {all | none | telnet | ssh}` со следующими параметрами:

`transport input all` или `transport input telnet ssh`. Поддерживать оба.
`transport input none`. Не поддерживать ни один.
`transport input telnet`. Поддерживать только Telnet.
`transport input ssh`. Поддерживать только SSH.

Шифрование и сокрытие паролей

В некоторых из команд конфигурации, рассмотренных в этой главе, упоминались пароли в виде открытого текста, сохраняющиеся в файле `running-config` (по крайней мере, изначально). Фактически только команда `enable secret` (из обсуждавшихся в этой главе до сих пор) автоматически скрывает значение пароля. Другие команды (команда `password` линий консоли и vty, а также пароль команды `username password`) оставляли пароль в виде обычного текста.

В следующих разделах обсуждаются возможности сокрытия значения пароля. Одни из них используют шифрование, другие — односторонний хеш-алгоритм. Независимо от деталей результатом будет невозможность просмотреть пароли в выводе команды `show running-config`.

Шифрование паролей при помощи команды `service password`

Чтобы предотвратить уязвимость пароля в отображаемой версии файла конфигурации или в его резервной копии, хранящейся на сервере, некоторые пароли можно зашифровать при помощи глобальной команды конфигурации `service password-encryption`. Эта команда влияет на способ хранения пароля команды `password` в режимах консоли и vty, а также глобальной команды `username password`. Правила команды `service password-config` приведены ниже.

- Немедленно после ввода команды `service password-encryption` операционная система IOS шифрует все существующие команды `password` (в режимах консоли и vty), а также пароли команды `username password`.
- Пока команда `service password-encryption` остается в конфигурации, операционная система IOS шифрует пароли, даже если их значения изменяются.
- Немедленно после ввода команды `no service password-encryption` шифрование паролей отменяется, но с существующими паролями операционная система IOS не делает ничего, оставляя их зашифрованными.
- После удаления команды `service password-encryption` из конфигурации операционная система IOS сохраняет значения всех измененных паролей этих команд в виде обычного текста.

Эти правила представлены в примере 8.5.

Пример 8.5. Шифрование и команда `service password-encryption`

```
Switch3# show running-config | begin line vty
line vty 0 4
  password cisco
  login
line vty 5 15
  password cisco
  login
```

```
Switch3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch3(config)# service password-encryption
Switch3(config)# ^Z

Switch3# show running-config | begin line vty
line vty 0 4
  password 7 070C285F4D06
  login
line vty 5 15
  password 7 070C285F4D06
  login
end
Switch3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch3(config)# no service password-encryption
Switch3(config)# ^Z
Switch3# show running-config | section vty
line vty 0 4
  password 7 070C285F4D06
  login
line vty 5 15
  password 7 070C285F4D06
  login
end

Switch3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch3(config)# line vty 0 4
Switch3(config-line)# password cisco
Switch3(config-line)# ^Z

Switch3# show running-config | begin line vty
line vty 0 4
  password cisco
  login
line vty 5 15
  password 7 070C285F4D06
  login
end
```

ВНИМАНИЕ!

Как уже упоминалось, тип шифрования "7", используемый командой `service password-encryption` для команд `password`, слаб. В Интернете вполне можно найти инструменты для расшифровки этих паролей. Фактически можно попробовать взять зашифрованный пароль из этого примера, найти соответствующий инструмент и расшифровать его как "cisco". Таким образом, команда `service password-encryption` отвадит любопытных, но не остановит подготовленного злоумышленника.

Пример 8.5 демонстрирует также несколько примеров применения символа | в конце команды CLI `show`. Символ | в конце команды `show` посылает вывод команды `begin` или `section` другой функции, как показано в примере 8.5. Функция `begin`, как демонстрирует команда `show running-config | begin line vty` в примере, получает

вывод от команды и отображает текст, начинающийся с первого вхождения заданного текста (в данном случае `line vty`). Возможности функции `| section vty` также представлены в примере 8.5 — отображается вывод только раздела о линиях vty.

Сокрытие привилегированного пароля

Коммутаторы могут защитить привилегированный режим, затребовав у пользователя ввод привилегированного пароля после ввода команды `enable`. Однако конфигурация может быть основана на двух разных командах: прежней глобальной команде `enable password` *пароль* и более новой глобальной команде `enable secret` *пароль*.

Операционная система IOS позволяет не использовать ни одну из этих команд, одну из них или обе. Далее следует выбрать пароль пользователя на основании следующих правил.

Ключевые факты о командах `enable secret` и `enable password`

- *Настроены обе команды.* Используйте команду `enable secret` *пароль*.
- *Настроена только одна команда.* Используйте пароль той же команды.
- *Ни одна из команд не настроена.* Пользователям консоли разрешен доступ к привилегированному режиму без ввода пароля, другим пользователям доступ запрещен.

Более новая команда `enable secret` обеспечивает лучшую защиту по сравнению с прежней командой `enable password`. Команда `enable password` сохраняет пароль как открытый текст, а единственной возможностью шифрования является довольно слабая команда `service password-encryption`. Более новая команда `enable secret` автоматически шифрует пароль, используя процесс, отличный от применяемого командой `service password-encryption`. Более новая команда применяет к паролю математическую функцию Message Digest 5 (MD5), сохраняя результат вычисления в файле конфигурации.

Пример 8.6 демонстрирует применение команды `enable secret` и описывает, как она скрывает текст пароля. Сначала в примере демонстрируется введенная пользователем команда `enable secret fred`. Затем команда `show running-configuration` демонстрирует, что операционная система IOS изменила команду `enable secret`, теперь применяется шифрование типа 5 (MD5). Непонятная длинная текстовая строка является хешем MD5 пароля, — человеку его не прочитать.

Пример 8.6. Шифрование и команда `enable secret`

```
Switch3(config)# enable secret ?
0       Specifies an UNENCRYPTED password will follow
5       Specifies an ENCRYPTED secret will follow
LINE    The UNENCRYPTED (cleartext) 'enable' secret
level   Set exec level password
Switch3(config)# enable secret fred
Switch3(config)# ^Z
Switch3# show running-config | include enable secret
```

```
enable secret 5 $1$ZGMA$e8cmvkz4UjiJhVp7.maLE1

Switch3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch3(config)# no enable secret
Switch3(config)# ^Z
```

В конце примера представлен также важный момент — отмена команды `enable secret`. Находясь в привилегированном режиме, вполне можно удалить пароль, используя команду `no enable secret`, даже без необходимости вводить значение пароля. Повторив команду, можно также переписать старый пароль.

ВНИМАНИЕ!

Пример 8.6 демонстрирует еще одно средство сокращения длинного вывода команды `show` при работе — команду `include`. Команда `show running-config | include enable secret` отображает только те строки вывода команды `show running-config`, которые включают текст (с учетом регистра) "`enable secret`".

И наконец, обратите внимание, что команда `enable secret` применила на маршрутизаторе Cisco другой хеш-алгоритм: SHA-256. Этот алгоритм мощней, чем MD5, операционная система IOS обозначает этот алгоритм как тип шифрования 4. Со временем компания Cisco, вероятно, добавит поддержку алгоритма SHA-256 и для коммутаторов. Результат использования алгоритмов SHA-256 и MD5 одинаков: пользователь настраивает такую команду, как `enable secret fred`, вводя открытый текстовый пароль, а операционная система IOS сохраняет его значение, зашифрованное алгоритмом MD5 (прежние версии IOS) или SHA-256 (более новые версии IOS).

Сокрытие паролей для локальных имен пользователя

Компания Cisco добавила команду `enable secret` для лучшей защиты паролем еще в 1990-х годах. Позже она добавила глобальную команду `username` *`пользователь`* `secret` *`пароль`* как альтернативу команде `username` *`пользователь`* `password` *`пароль`*. Эта команда использует алгоритм шифрования SHA-256 (тип 4).

Ныне команда `username secret` предпочтительней команды `username password`, как и команда `enable_secret` предпочтительней команды `enable_password`. Однако имя пользователя может быть задано командой `username secret` или `username password`, но не обеими.

Настройки консольного соединения и канала vty

В этом разделе описаны некоторые второстепенные настройки, влияющие на поведение интерфейса командной строки (CLI), и соединения через консоль и/или сеансы vty (Telnet и SSH).

Сообщения, отображаемые при подключении

Коммутаторы компании Cisco могут отображать несколько различных сообщений при подключении к ним, в зависимости от того, что задали системные администраторы. Отображаемое при подключении *сообщение* (banner) — это просто текст, который выводится на экран пользователя. Можно настроить маршрутизатор или коммутатор так, что он будет показывать несколько сообщений, например, какой-

то текст до аутентификации в системе, а какой-то после. В табл. 8.1 приведены самые популярные сообщения и примеры их типичного применения.

Команда `banner` режима глобальной конфигурации применяется для настройки трех следующих типов сообщений.

Таблица 8.1. Сообщения, отображаемые при подключении

Сообщение	Пример использования
Сообщение дня (Message of the Day — MOTD)	Отображается до того, как появится приглашение аутентификации. Используется для некоторых временных сообщений, например, "Маршрутизатор будет выключен в полночь для технического осмотра"[6]
Сообщение перед аутентификацией (login)	Отображается до выполнения аутентификации, но после сообщения дня. Обычно используется для какого-либо постоянного сообщения, например: "Неавторизованный доступ к устройству запрещен"
Сообщение после аутентификации (exec)	Отображается после успешной аутентификации пользователя. Обычно используется для вывода информации, которая должна быть скрыта от неавторизованных пользователей

Формат команды универсален — в качестве первого параметра указывается тип сообщения; если же параметр не указан, операционная система стандартно использует его как сообщение MOTD. Текст сообщения может состоять из нескольких строк, для этого достаточно нажать клавишу <Enter> и продолжить ввод текста. Интерфейс командной строки обнаруживает, что ввод сообщения закончен, когда пользователь вводит тот же разделительный символ, который он вводил в начале текста.

В примере 8.7 показан процесс конфигурирования трех сообщений, перечисленных в табл. 8.1, а также процесс подключения пользователя. Первое сообщение, сообщение дня (MOTD), в конфигурации приведено без ключевого слова `motd`, поскольку обычно, если не указаны параметры, операционная система привязывает текст к нему. В первых двух вариантах команды используется разделительный символ `#`. В третьем варианте используется символ `Z`, чтобы проиллюстрировать, что можно использовать разные символы, главное, чтобы начальный и конечный символы были одинаковыми. Кроме того, в последнем варианте команды `banner` вводится несколько строк текста.

Пример 8.7. Настройка сообщений

```
! Ниже показан пример настройки отображаемых при аутентификации
! сообщений.
! Обратите внимание, что любой символ можно использовать в качестве
! разделителя, главное, чтобы он не был частью вводимого текста.


SW1(config)# banner #
Enter TEXT message. End with the character '#'.
(MOTD) Switch down for maintenance at 11PM Today #
```

[6] Как и все команды конфигурации, описания интерфейсов, комментарии, такие сообщения можно писать только латиницей. — *Примеч. ред.*

```
SW1(config)# banner login #
Enter TEXT message. End with the character '#'.
(Login) Unauthorized Access Prohibited!!!!
#
SW1(config)# banner exec Z
Enter TEXT message. End with the character 'Z'.
(Exec) Company picnic at the park on Saturday
Don't tell outsiders!
Z
SW1(config)# end


! Ниже показано, что пользователь сначала отключается от устройства,
! потом подключается заново.
! Он видит сообщение дня, потом сообщение аутентификации, приглашение
! операционной системы и еще одно сообщение.
SW1# quit

SW1 con0 is now available

Press RETURN to get started.

(MOTD) Switch down for maintenance at 11PM Today
(Login) Unauthorized Access Prohibited!!!!

User Access Verification

Username: fred
Password:
(Exec) Company picnic at the park on Saturday
don't tell outsiders!
SW1>
```

Буфер истории команд

Когда пользователь вводит несколько команд подряд в интерфейсе командной строки (CLI), несколько последних команд сохраняются в буфере истории команд. Как было сказано в главе 7, можно использовать клавишу <↑> или комбинацию клавиш <Ctrl+P>, чтобы переместиться вверх по истории команд и вызвать одну из предыдущих настроек. Эта функция значительно облегчает процесс ввода повторяющихся или подобных команд. В табл. 8.2 перечислены некоторые из наиболее полезных команд для работы с буфером истории команд.

Ключевая тема **Таблица 8.2. Команды буфера истории команд**

Команда	Описание
`show history`	Отображает команды, находящиеся в буфере истории команд
`history size` *x*	Вводится в режиме конфигурации консольного подключения или сеанса vty и задает количество команд (*x*), которое будет сохраняться в буфере истории команд, соответственно для консольного или сеанса vty
`terminal history size` *x*	Позволяет задавать размер буфера истории команд (*x*) только для текущего сеанса пользователя

Команды logging synchronous и exec-timeout

В следующем разделе кратко рассматривается несколько способов упрощения работы с консолью, чтобы коммутатор не прервал его регистрационными сообщениями, а также продолжительность соединения с консолью прежде, чем оно будет разорвано.

Консоль автоматически получает копии всех сообщений системного журнала на коммутаторе. Идея в том, что если коммутатор должен сообщить сетевому администратору некую важную, а возможно и срочную информацию, то на консоли он мог бы увидеть ее сразу.

Глобальные команды `no logging console` и `logging console` позволяют запретить и разрешить отображение этих сообщений на консоли. Например, если сообщения регистрации мешают работе с консолью и их отображение следует временно отключить, воспользуйтесь глобальной командой конфигурации `no logging console`, а затем можно снова разрешить.

К сожалению, изначально операционная система IOS выводит сообщения системного журнала на экран консоли в любой момент времени, вполне возможно, что и прямо при вводе команды или посередине вывода команды `show`. Внезапное появление неожиданных строк текста может немного раздражать.

Операционная система IOS предоставляет решение этой проблемы, позволяя отображать сообщения системного журнала только в более удобные моменты, например в конце вывода команды `show`. Для этого следует ввести подкоманду канала консоли `logging synchronous`.

Еще один способ повышения удобства работы с консолью подразумевает контроль периода ее работы. Изначально коммутатор автоматически отключает пользователей консоли и линий vty (Telnet и SSH) после 5 минут бездеятельности. Подкоманда канала `exec-timeout` *`минут секунд`* позволяет изменить этот период или предотвратить отключение вообще, введя значения 0 минут и 0 секунд.

Синтаксис этих двух команд на канале консоли демонстрирует пример 8.8. Обратите внимание, что обе команды применимы также к линиям vty по тем же причинам.

Пример 8.8. Настройка периодов бездеятельности консоли и моментов отображения регистрационных сообщений

```
line console 0
  login
  password cisco
  exec-timeout 0 0
  logging synchronous
```

ВНИМАНИЕ!

На этом завершается первая часть этой главы. Если вы еще не опробовали команды на маршрутизаторе или коммутаторе, то сейчас подходящий момент сделать паузу в чтении и опробовать некоторые из них. Если у вас есть реальные устройства или эмулятор Pearson Simulator, выполните лабораторные работы на переходы в интерфейсе CLI, установку паролей и другие простые административные действия. В противном случае посмотрите видеофильмы с образа DVD-диска по навигации в CLI и настройке конфигурации. Кроме того, опробуйте несколько лабораторных работ на эмуляторе ICND1 Simulator Lite, также имеющемся на образе этого DVD-диска. Это позволит получить лучшее представление об обращении с интерфейсом командной строки.

Настройка коммутаторов локальных сетей и управление ими

Коммутатор Cisco очень хорошо работает и с фабричной конфигурацией без всякой настройки. Они поставляются со стандартными настройками, при всех разрешенных интерфейсах (стандартное состояние `no shutdown`) и с автопереговорами для всех портов, которые могут их использовать (стандартное состояние `duplex auto` и `speed auto`). Все интерфейсы стандартно являются частью сети VLAN 1 (`switchport access vlan 1`). Достаточно подключить к новому коммутатору Cisco все физические кабели Ethernet, включить питание, и коммутатор готов к работе.

В большинстве корпоративных сетей коммутатор должен работать с некоторыми параметрами, отличными от заводских. Далее в главе обсуждаются некоторые из этих параметров, а их более подробное описание приведено в главе 9. (Подробности в этом разделе отличаются от таковых для маршрутизатора.) В частности, в настоящем разделе рассматривается следующее:

- настройка IP-адреса для дистанционного доступа;
- настройка интерфейса (включая скорость и дуплекс);
- настройка защиты порта;
- защита неиспользованных интерфейсов коммутатора.

Предоставление IP-адреса для дистанционного доступа

Чтобы позволить клиенту Telnet или SSH обращаться к коммутатору и позволить это другим протоколам управления (например, *простому протоколу управления сетью* (Simple Network Management Protocol — SNMP)), основанным на протоколе IP, коммутатор нуждается в IP-адресе. IP-адрес не имеет никакого отношения к тому, как коммутаторы перенаправляют фреймы Ethernet; он нужен для поддержки служебного управляющего трафика.

Настройка IP-адреса коммутатора подобна таковой у компьютера с одним интерфейсом Ethernet. С этой точки зрения у компьютера есть процессор, выполняющий операционную систему. У него есть плата сетевого интерфейса Ethernet (сетевая плата). Согласно конфигурации операционной системы, с сетевой платой связан IP-адрес, заданный вручную или полученный динамически от сервера DHCP.

Коммутатор использует концепции, подобные хосту, за исключением того, что он может использовать виртуальную сетевую плату. Как и у компьютера, у коммутатора есть процессор, выполняющий операционную систему (IOS). Коммутатор использует концепцию, подобную сетевой плате, — *коммутируемый виртуальный интерфейс* (Switched Virtual Interface — SVI) или более привычно — *интерфейс VLAN* (VLAN interface), действующий как собственная сетевая плата коммутатора для подключения к локальной сети и передачи пакетов IP. Подобно хосту, настройка коммутатора подразумевает установку таких параметров IP, как IP-адрес для этого интерфейса VLAN (рис. 8.5).

Типичный коммутатор LAN Cisco уровня 2 может использовать только один интерфейс VLAN, но какой конкретно, выбирает сетевой инженер, перемещая управляющий трафик коммутатора на специфический интерфейс. Например, на рис. 8.6 показан коммутатор с несколькими физическими портами для двух разных интер-

фейсов VLAN (1 и 2). Сетевой инженер должен выбрать, какой из IP-адресов коммутатора используется для доступа к коммутатору, и иметь IP-адрес в подсети 192.168.1.0 (VLAN 1) или 192.168.2.0 (VLAN 2).

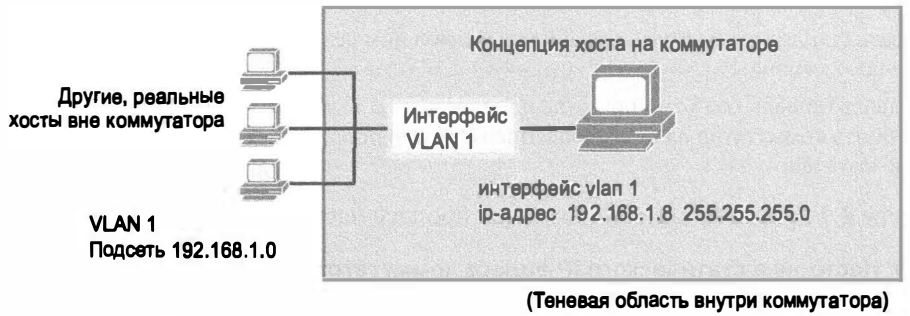


Рис. 8.5. Концепция виртуального интерфейса коммутатора (SVI)

Рис. 8.6. Выбор VLAN для настройки IP-адреса коммутатора

ВНИМАНИЕ!

Некоторые коммутаторы Cisco называют *коммутатором уровня 2* (Layer 2 switch), они перенаправляют фреймы Ethernet так, как описано в главе 6. Другие коммутаторы Cisco называют *многоуровневыми коммутаторами* (multilayer switch) или *коммутаторами третьего уровня* (Layer 3 switch), они способны перенаправлять пакеты IP, используя логику уровня 3, обычно используемую маршрутизаторами. Коммутаторы третьего уровня настраивают IP-адреса на нескольких интерфейсах VLAN одновременно. В этой главе подразумевается, что все коммутаторы принадлежат уровню 2. Более подробная информация о различиях между этими типами коммутаторов LAN приведена в главе 9.

Настройка IPv4-адреса на коммутаторе

Коммутатор настраивает свой IPv4-адрес и маску на специальном, подобном сетевой плате, *интерфейсе VLAN*. Ниже приведена последовательность действий по настройке IPv4-адреса на интерфейсе VLAN 1 коммутатора.

Настройка IP-адреса и стандартного шлюза коммутатора

Ключевая тема

Этап 1 Перейти в режим конфигурации сети VLAN 1 с помощью команды `interface vlan 1` из глобального режима конфигурации устройства

Этап 2 Присвоить IP-адрес и маску с помощью команды `ip address` *ip-адрес маска* в подрежиме конфигурации интерфейса

Этап 3 Включить виртуальный интерфейс сети VLAN 1 с помощью команды `no shutdown` в подрежиме конфигурации интерфейса

Этап 4 Указать стандартный шлюз устройства в глобальном режиме конфигурации с помощью команды `ip default-gateway` *ip-адрес*

Этап 5 Добавить глобальную команду `ip name-server` *ip-адрес1 ip-адрес2...*, чтобы настроить коммутатор на использование DNS при поиске имен по их IP-адресам (необязательно)

В примере 8.9 приведен описанный выше процесс настройки адреса.

Пример 8.9. Настройка статического IP-адреса коммутатора

```
Emma# configure terminal
Emma(config)# interface vlan 1
Emma(config-if)# ip address 192.168.1.200 255.255.255.0
Emma(config-if)# no shutdown
00:25:07: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:25:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
  state to up
Emma(config-if)# exit
Emma(config)# ip default-gateway 192.168.1.1
```

Обратите внимание на особенно важную команду: `[no] shutdown`. Чтобы административно включить интерфейс на коммутаторе, используйте подкоманду интерфейса `no shutdown`, а чтобы отключить его — подкоманду интерфейса `shutdown`. Сообщения, представленные в примере 8.9, непосредственно после команды `no shutdown` являются созданными коммутатором сообщениями системного журнала, оповещающими о включении интерфейса.

Коммутатор может также использовать сервер DHCP для динамического получения параметров протокола IPv4. Все, что для этого нужно, — указать коммутатору использовать протокол DHCP на интерфейсе и включить интерфейс. С учетом, что сервер DHCP работает в этой сети, коммутатор сам изучит все параметры. Список этапов настройки интерфейса VLAN 1 приведен ниже.

Ключевая тема

Настройка конфигурации коммутатора на изучение параметров IP в режиме клиента DHCP

Этап 1 Войдите в режим конфигурации интерфейса VLAN 1, используя глобальную команду конфигурации `interface vlan 1`, и при необходимости включите интерфейс командой `no shutdown`

Этап 2 Присвойте IP-адрес и маску, используя подкоманду интерфейса `ip address dhcp`

Пример 8.10 иллюстрирует настройку устройства с использованием протокола DHCP.

Пример 8.10. Динамическая установка IP-адреса с помощью протокола DHCP

```
Emma# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Emma(config)# interface vlan 1
Emma(config-if)# ip address dhcp
Emma(config-if)# no shutdown
Emma(config-if)# ^Z
Emma#
00:38:20: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:38:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
state to up
```

Проверка IPv4-адреса на коммутаторе

Конфигурацию IPv4-адреса коммутатора можно проверить несколькими способами. Во-первых, всегда можно просмотреть текущую конфигурацию, используя команду `show running-config`. Во-вторых, можно просмотреть информацию об IP-адресе и маске, используя команду `show interface vlan` *x*, выводящую подробную информацию о состоянии интерфейса VLAN *x*. И наконец, при использовании сервера DHCP команда `show dhcp lease` позволяет просмотреть зарезервированный (временно) IP-адрес и другие параметры. (Коммутатор не сохраняет конфигурацию IP, полученную от сервера DHCP в файле `running-config`.) Типичный вывод этих команд, в соответствии с конфигурацией примера 8.10, приведен в примере 8.11.

Пример 8.11. Проверка информации, полученной коммутатором от сервера DHCP

```
Emma# show dhcp lease
Temp IP addr: 192.168.1.101 for peer on Interface: Vlan1
Temp sub net mask: 255.255.255.0
    DHCP Lease server: 192.168.1.1, state: 3 Bound
    DHCP transaction id: 1966
    Lease: 86400 secs, Renewal: 43200 secs, Rebind: 75600 secs
Temp default-gateway addr: 192.168.1.1
    Next timer fires after: 11:59:45
    Retry count: 0 Client-ID: cisco-0019.e86a.6fc0-Vl1
    Hostname: Emma
Emma# show interface vlan 1
Vlan1 is up, line protocol is up
    Hardware is EtherSVI, address is 0019.e86a.6fc0 (bia 0019.e86a.6fc0)
    Internet address is 192.168.1.101/24
    MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
        reliability 255/255, txload 1/255, rxload 1/255
! Остальная информация опущена
Emma# show ip default-gateway
192.168.1.1
```

В последней части примера 8.11 показан результат выполнения команды `show interface vlan 1`, в котором выделены две наиболее важные строки, связанные с адресом коммутатора. В первой выделенной строке отображается состояние интерфейса VLAN 1, в данном случае это "`up and up`" (активен и активен). Если интерфейс виртуальной локальной сети неактивен, коммутатор не может отправлять и принимать трафик для своего IP-адреса. Такая ситуация может быть в том случае, если системный администратор забыл ввести команду `no shutdown`, а виртуальный интерфейс стандартно находится в выключенном состоянии и в результате вышеуказанной команды будет отображаться надпись "`administratively down`" (административно выключен).

Обратите внимание на IP-адрес интерфейса в третьей строке вывода. Если задать IP-адрес статически, как в примере 8.9, он будет отображаться всегда. Но если используется сервер DHCP и сервер DHCP откажет, то команда `show interfaces vlan x` не отобразит здесь IP-адрес. Когда сервер DHCP работает, эта команда позволяет просмотреть IP-адрес, но не выяснить, задан ли он статически или получен на время от сервера DHCP.

Настройка интерфейсов коммутатора

В операционной системе Cisco IOS термин *интерфейс* (interface) используется для обозначения любого физического *порта* (port), пересылающего данные от одних устройств другим. Обычно оба термина используются как синонимы и в дальнейшем так и будут использоваться во всей книге. Для интерфейсов могут быть заданы разные независимые параметры.

В операционной системе Cisco IOS для настройки интерфейсов используется специализированный режим конфигурирования интерфейса, называемый обычно подрежимом интерфейса. Например, в таком подрежиме могут быть использованы команды `duplex` и `speed` для статического указания настроек порта или может использоваться автоматическое определение скорости и дуплексного режима (это стандартная настройка). В примере 8.12 показаны процесс настройки дуплексного режима и скорости, а также использование команды `description` (описание), которая задает некоторое текстовое описание интерфейса, чтобы было понятно его назначение.

Пример 8.12. Базовые настройки интерфейса

```
Emma# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Emma(config)# interface FastEthernet 0/1
Emma(config-if)# duplex full
Emma(config-if)# speed 100
Emma(config-if)# description Server1 connects here
Emma(config-if)# exit
Emma(config)# interface range FastEthernet 0/11 - 20
Emma(config-if-range)# description end-users connect_here
Emma(config-if-range)# ^Z
Emma#
Emma# show interfaces status
Port  Name               Status       Vlan Duplex  Speed  Type
Fa0/1 Server1 connects h notconnect  1    full    100    10/100BaseTX
Fa0/2                    notconnect  1    auto    auto   10/100BaseTX
Fa0/3                    notconnect  1    auto    auto   10/100BaseTX
Fa0/4                    connected   1    a-full  a-100  10/100BaseTX
Fa0/5                    notconnect  1    auto    auto   10/100BaseTX
Fa0/6                    connected   1    a-full  a-100  10/100BaseTX
Fa0/7                    notconnect  1    auto    auto   10/100BaseTX
Fa0/8                    notconnect  1    auto    auto   10/100BaseTX
Fa0/9                    notconnect  1    auto    auto   10/100BaseTX
Fa0/10                   notconnect  1    auto    auto   10/100BaseTX
Fa0/11 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/12 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/13 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/14 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/15 end-users connect notconnect  1    auto    auto   10/100BaseTX
```

```
Fa0/16 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/17 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/18 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/19 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/20 end-users connect notconnect  1    auto    auto   10/100BaseTX
Fa0/21                   notconnect  1    auto    auto   10/100BaseTX
Fa0/22                   notconnect  1    auto    auto   10/100BaseTX
Fa0/23                   notconnect  1    auto    auto   10/100BaseTX
Fa0/24                   notconnect  1    auto    auto   10/100BaseTX
Gi0/1                    notconnect  1    auto    auto  10/100/1000BaseTX
Gi0/2                    notconnect  1    auto    auto  10/100/1000BaseTX
```

Как именно настроены интерфейсы, можно узнать с помощью команды `show running-config` (она в примере не показана), а краткую информацию можно просмотреть с помощью команды `show interfaces status`. Результат выполнения последней — короткая строка информации о каждом интерфейсе, в первой части которой есть часть описания интерфейса (если оно введено), а во второй отображаются параметры дуплексного режима и скорости. Несколько из приведенных выше строк вывода преднамеренно демонстрируют некоторые различия.

`FastEthernet 0/1 (Fa0/1)`. Вывод отображает заданную скорость 100 и полный дуплекс; однако состояние порта указано как `notconnect` (не подключен). Это значит, что физический канал связи в настоящее время не работает, возможно, к нему не подключен кабель, или выключено устройство на его противоположном конце, или отключен порт противоположного устройства. В данном случае к порту не был подключен кабель.

`FastEthernet 0/2 (Fa0/2)`. К этому порту также не подключен кабель, но он использует стандартную конфигурацию. Выделенные части вывода демонстрируют стандартные параметры интерфейса, означающие автопереговоры.

`FastEthernet 0/4 (Fa0/4)`. Подобно порту Fa0/2, этот порт имеет стандартную конфигурацию, но он подключен к работающему устройству, поэтому его состояние указано как `connected` (подключен). Это устройство уже завершило процесс автопереговоров, вывод демонстрирует согласованную скорость и дуплекс (`a-full` и `a-100`), приставка `a-` свидетельствует о том факте, что эти значения получены в результате автопереговоров.

Следует также обратить внимание на то, что в операционной системе для удобной и более эффективной работы есть возможность настраивать сразу диапазон интерфейсов с помощью команды `interface range`. В приведенном выше примере команда `interface range FastEthernet 0/11 - 20` инструктирует операционную систему IOS о том, что все последующие команды будут применяться к интерфейсам с Fa0/11 по Fa0/20.

ВНИМАНИЕ!

Настройка скорости и дуплекса на интерфейсе коммутатора Cisco отключает автопереговоры.

Защита портов

Если сетевой инженер точно знает, какие конкретно устройства будут подключены кабелями к каким интерфейсам коммутатора, он может использовать *защиту порта* (port security), чтобы его могли использовать только указанные устройства.

Это затрудняет злоумышленнику возможность подключить портативный компьютер к неиспользуемому порту коммутатора. Когда недозволенное устройство пытается посылать фреймы на интерфейс коммутатора, он может предпринять в ответ различные действия: от вывода информационных сообщений до отключения интерфейса.

Защита порта идентифицирует устройства по MAC-адресу отправителя во фрейме Ethernet. Например, компьютер PC1 на рис. 8.7 посылает фрейм с MAC-адресом отправителя, принадлежащим компьютеру PC1. На интерфейсе F0/1 коммутатора SW1 может быть настроена защита порта, и если это так, то коммутатор SW1, зная MAC-адрес компьютера PC1, позволит передать его фреймы через порт F0/1.

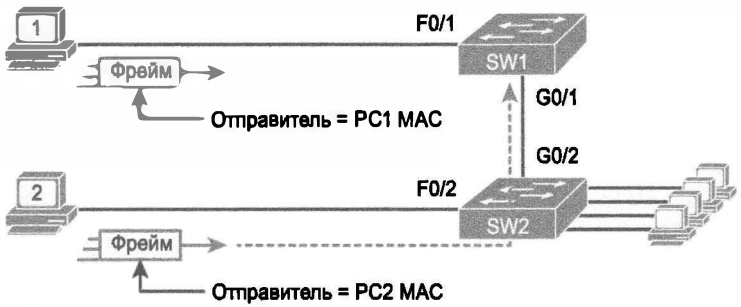


Рис. 8.7. MAC-адреса отправителя во фреймах, поступающих на коммутатор

Защита порта не делает различий между фреймами, поступившими от локального устройства и перенаправленными через другие коммутаторы. Например, коммутатор SW1 мог использовать защиту порта G0/1, проверяя MAC-адрес отправителя фрейма, посланного компьютером PC2 через коммутатор SW2 на коммутатор SW1.

У защиты порта много возможностей, но основные принципы работы у них одинаковы. Коммутаторы устанавливают защиту на каждый порт индивидуально, для каждого порта допустимы свои параметры. Каждый порт имеет максимально допустимое количество разрешенных MAC-адресов, т.е. для всех вводящих на этот порт фреймов допустимо только это количество *разных* MAC-адресов отправителя. Когда поступает фрейм с новым MAC-адресом отправителя, при его превышении защита порта считает, что происходит попытка взлома. В этом случае коммутатор стандартно отказывается от всего последующего входящего трафика на данном порту.

Все варианты защиты порта кратко описаны ниже.

Ключевая тема

Основные варианты защиты порта

- Определите максимальное разрешенное количество MAC-адресов отправителя для всех входящих фреймов на интерфейс.
- Отследите все входящие фреймы и сохраните список всех MAC-адресов отправителей, добавьте счетчик количества отличных MAC-адресов отправителя.
- Если при добавлении нового MAC-адреса отправителя в список количество хранимых MAC-адресов превысит заданный максимум, срабатывает защита порта и коммутатор принимает меры (стандартное действие — отключение интерфейса).

Эти правила определяют основные методы защита порта, но есть другие возможности, включая настройку определенных MAC-адресов, которым позволено посылать фреймы на интерфейс. Например, на рис. 8.7 коммутатор SW1 подключен через интерфейс F0/1 к компьютеру PC1, поэтому конфигурация защиты порта могла бы включить MAC-адрес компьютера PC1 в список допустимых MAC-адресов. Но предварительное определение MAC-адресов не является обязательным для защиты порта: можно определить все MAC-адреса, ни один из них или некое подмножество.

Весьма соблазнительна идея предопределить все MAC-адреса для защиты порта, но найти MAC-адрес каждого устройства может быть довольно хлопотной. Защита порта предоставляет простой способ обнаружения всех MAC-адресов, используемых с каждым портом, — это *автоматическое обнаружение MAC-адресов* (sticky secure MAC addresses). Защита порта сама изучает MAC-адреса на каждом порту и сохраняет их в конфигурации (в файле `running-config`). Это позволяет существенно сократить усилия по обнаружению MAC-адресов каждого устройства.

Как можно заметить, у защиты порта есть много подобных возможностей. Они рассматриваются в нескольких следующих разделах.

Настройка защиты порта

Настройка защиты порта подразумевает несколько этапов. Сначала необходимо отключить автопереговоры, которое не обсуждаются до главы 9, будь то порт магистрального канала или порт доступа. На настоящий момент достаточно знать, что защита порта требует настроить порт как порт доступа или магистральный порт. Остальная часть команд защиты порта позволяет установить максимально допустимое количество MAC-адресов на порт и настроить фактические MAC-адреса, как описано в следующей последовательности.

Последовательность настройки защиты порта

Ключевая тема

Этап 1 Используя подкоманды интерфейса `switchport mode access` или `switchport mode trunk`, объявите интерфейс коммутатора статическим портом доступа или магистральным портом соответственно

Этап 2 Включите защиту порта подкомандой интерфейса `switchport port-security`

Этап 3 Переопределите стандартное максимальное количество позволенных MAC-адресов, интерфейса (в данном случае1) подкомандой интерфейса `switchport port-security maximum` *число* (Необязательно.)

Этап 4 Переопределите стандартное действие (отключение), предпринимаемое при нарушении защиты. Используйте подкоманду интерфейса `switchport port-security violation {protect | restrict | shutdown}` (Необязательно.)

Этап 5 Задайте все допустимые MAC-адреса отправителей для данного интерфейса, используя команду `switchport port-security mac-address` *MAC-адрес*. Чтобы задать несколько MAC-адресов, используйте эту команду многократно (Необязательно.)

Этап 6 Можно также включить автоматическое обнаружение MAC-адресов, чтобы коммутатор сам изучил MAC-адреса. Используйте подкоманду интерфейса `switchport port-security mac-address sticky` (Необязательно.)

На рис. 8.8 и в примере 8.13 показаны четыре примера защиты порта, каждый с различными параметрами для демонстрации разных возможностей.

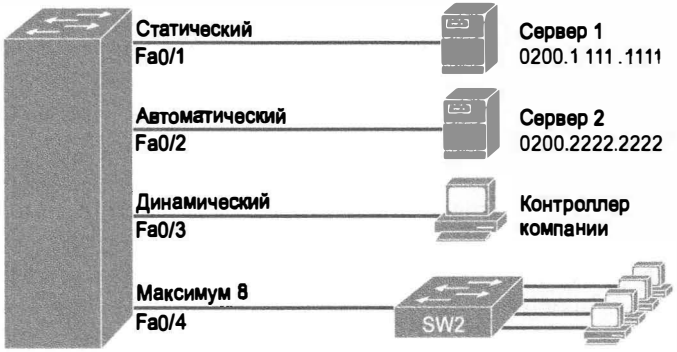


Рис. 8.8. Пример настройки защиты порта

Пример 8.13. Разновидности настройки защиты порта

```
fred# show running-config
! (Часть конфигурации опущена для краткости)

interface FastEthernet0/1
  switchport mode access
  switchport port-security
  switchport port-security mac-address 0200.1111.1111
!
interface FastEthernet0/2
  switchport mode access
  switchport port-security
  switchport port-security mac-address sticky
!
interface FastEthernet0/3
  switchport mode access
  switchport port-security
!
interface FastEthernet0/4
  switchport mode access
  switchport port-security
  switchport port-security maximum 8
```

Сначала рассмотрим конфигурацию всех четырех интерфейсов в примере 8.13, сосредоточившись на первых двух подкомандах интерфейса. Обратите внимание, что все четыре интерфейса в примере используют те же первые две подкоманды интерфейса, соответствующие первым двум этапам настройки. Команда `switchport port-security` включает защиту порта со стандартными значениями, а команда `switchport mode access` выполняет требование, согласно которому настраиваемый порт должен быть либо портом доступа, либо портом магистрального канала.

Теперь снова рассмотрим все четыре интерфейса и обратим внимание на то, что после первых двух подкоманд интерфейса конфигурация у каждого интерфейса разная. Просто каждый интерфейс демонстрирует свой пример настройки.

Первый интерфейс, FastEthernet 0/1, использует одну дополнительную подкоманду защиты порта: `switchport port-security mac-address 0200.1111.1111`, которая определяет конкретный MAC-адрес отправителя. При стандартном максимальном количестве адресов отправителя 1 на этом порту будут позволены только

фреймы с MAC-адресом отправителя 0200.1111.1111. Когда на порт F0/1 поступит фрейм с MAC-адресом отправителя, отличным от 0200.1111.1111, коммутатор предпримет стандартное действие — отключит интерфейс.

Второй интерфейс, FastEthernet 0/2, использует ту же логику, что и первый, но использует автоматическое обнаружение MAC-адресов, задаваемое командой `switchport port-security mac-address sticky`. Конец следующего примера 8.14 демонстрирует файл конфигурации `running-config`, отображающий автоматическое обнаружение MAC-адресов в данном случае.

ВНИМАНИЕ!

Защита порта не сохраняет конфигурацию автоматического обнаружения. По мере необходимости используйте команду `copy running-config startup-config`.

Два других интерфейса не используют ни предопределение, ни автоматическое обнаружение MAC-адресов. Единственное различие между конфигурациями защиты порта этих двух интерфейсов в том, что порт FastEthernet 0/4 допускает восемь MAC-адресов, поскольку он подключен к другим коммутаторам и должен получить фреймы с MAC-адресами нескольких отправителей. Интерфейс F0/3 использует стандартное значение — максимум один MAC-адрес.

Проверка защиты порта

В примере 8.14 приведен вывод двух команд `show port-security interface`. Они отображают конфигурации защиты портов, а также несколько важных фактов о текущей работе зашиты порта, включая информацию о любых нарушениях безопасности. Обе команды в примере относятся к интерфейсам F0/1 и F0/2, настроенным в примере 8.13.

Пример 8.14. Использование защиты порта для определения допустимых MAC-адресов отдельных интерфейсов

```
SW1# show port-security interface fastEthernet 0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0013.197b.5004:1
Security Violation Count   : 1
SW1# show port-security interface fastEthernet 0/2
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
```

```
Configured MAC Addresses      : 1
Sticky MAC Addresses          : 1
Last Source Address:Vlan      : 0200.2222.2222:1
Security Violation Count      : 0
SW1# show running-config
(строки опущены для краткости)
interface FastEthernet0/2
  switchport mode access
  switchport port-security
  switchport port-security mac-address sticky
  switchport port-security mac-address sticky 0200.2222.2222
```

Вывод первых двух команд в примере 8.14 подтверждает, что на интерфейсе FastEthernet 0/1 произошло нарушение безопасности, а на интерфейсе FastEthernet 0/2 — нет. Вывод команды `show portsecurity interface fastethernet 0/1` демонстрирует, что интерфейс находится в состоянии `secure-shutdown` (защищен–отключен), т.е. был отключен из-за нарушения защиты порта. В данном случае нарушение вызвало другое устройство, подключенное к порту F0/1, послав фрейм с MAC-адресом отправителя, отличным от 0200.1111.1111. Но порт Fa0/2, использующий автоматическое обнаружение, просто изучил MAC-адрес, используемый сервером Server 2.

Внизу в примере 8.14 представлены изменения по сравнению с конфигурацией в примере 8.13, сохраненные в конфигурационном файле `running-config` устройства, вызванные подкомандой интерфейса `switchport port-security mac-address sticky 0200.2222.2222`.

Действия защиты порта

И наконец, коммутатор может быть настроен так, чтобы использовать при нарушении безопасности одно из трех действий. Все три возможности подразумевают отказ коммутатора от подозрительных фреймов, но в некоторых случаях коммутатор предпринимает дополнительные действия: передачу сообщений системного журнала на консоль, передачу сообщений SNMP на станцию управления сетью и отключение интерфейса. В табл. 8.3 перечислены некоторые из возможностей команды `switchport port-security violation {protect | restrict | shutdown}` и их значение.

Ключевая тема **Таблица 8.3. Действия при нарушении защиты порта**

Параметр/Действие при нарушении безопасности	Защита (`protect`)	Ограничение (`restrict`)	Выключение (`shutdown`, стандартное действие)
Отбрасывание подозрительного трафика	Да	Да	Да
Отправка сообщения в системный журнал и через протокол SNMP	Нет	Да	Да
Выключение интерфейса и блокирование всего трафика	Нет	Нет	Да

Обратите внимание: возможность отключения не добавляет фактически подкоманду `shutdown` в конфигурацию интерфейса. Вместо этого операционная система IOS переводит интерфейс в состояние `error disabled` (отключено из-за ошибки), подразумевающее прекращение передачи коммутатором всех входящих и исходящих фреймов. Для выхода из этого состояния кто-то должен вручную отключить интерфейс командой `shutdown`, а затем снова включить командой `no shutdown`.

Защита неиспользуемых интерфейсов коммутатора

Компания Cisco разработала стандартные настройки интерфейсов коммутаторов таким образом, чтобы они работали сразу после включения устройства без какой-либо дополнительной конфигурации. Однако у стандартных значений есть существенный побочный эффект — ухудшение защиты. Они автоматически согласовывают скорость и дуплекс, а каждый интерфейс находится во включенном состоянии (`no shutdown`), все они привязаны к сети VLAN 1 и т.д. Поэтому интерфейсы в стандартной конфигурации могут быть использованы злоумышленником для доступа к локальной сети. Компания Cisco предлагает следующий набор общих рекомендаций для переопределения стандартных параметров интерфейса и защиты неиспользуемых портов.

Рекомендованные настройки неиспользуемых портов коммутатора

Ключевая тема

- административно отключить интерфейсы с помощью команды `shutdown` подрежима конфигурации интерфейса;
- отключить автоматическое согласование магистрального соединения (trunking) и протокола VTP, переведя порт в режим работы доступа к сети (access) с помощью команды `switchport mode access` подрежима конфигурации интерфейса;
- привязать все неиспользуемые порты к неиспользуемой сети VLAN с помощью команды `switchport access vlan` *номер* подрежима конфигурации интерфейса.

Откровенно говоря, административное отключение интерфейса является самым надежным методом и закрывает все бреши в защите устройства. Тем не менее два последних действия помогут избежать множества проблем в том случае, если кто-либо перенастроит устройство и включит интерфейс с помощью команды `no shudown`.

Обзор

Резюме

- Первый этап зашиты коммутатора — это защита доступа к интерфейсу CLI. Она подразумевает зашиту доступа к привилегированному режиму, поскольку именно из него злоумышленник может перезагрузить коммутатор или изменить его конфигурацию.
- Коммутаторы Cisco защищают привилегированный режим при помощи привилегированного пароля. Пользователь в пользовательском режиме вводит команду `enable`, запрашивающую привилегированный пароль; если пользователь вводит правильный пароль, операционная система IOS переводит пользователя в привилегированный режим.
- Настройка паролей консоли и vty использует те же две подкоманды для консоли и vty соответственно, в режиме конфигурации линии. Команда `login` указывает операционной системе IOS использовать простой пароль, а команда `password` *`пароль_значение`* задает пароль. Операционная система IOS защищает привилегированный режим, используя привилегированный пароль, заданный глобальной командой `enable secret` *`пароль_значение`*.
- Переход от доступа только по паролю к локально заданным именам пользователя и паролям требует лишь немногих изменений в конфигурации. Для этого достаточно одной или нескольких глобальных команд конфигурации `username` *имя* `password` *пароль*.
- Коммутаторы и маршрутизаторы Cisco поддерживают еще один способ проверки правильности имен пользователя и паролей — внешний сервер AAA. При этом подходе аутентификации коммутатор (или маршрутизатор) просто посылает на сервер AAA сообщение с запросом, допустимо ли данное имя пользователя и пароль, а сервер AAA отвечает.
- Для поддержки коммутаторами Cisco протокола SSH требуется не только базовая конфигурация с применением имен пользователей и паролей Telnet, но и дополнительная. Изначально на коммутаторе уже выполняется сервер SSH, он принимает входящие соединения от клиентов SSH. Кроме того, коммутатор нуждается в криптографическом ключе (cryptography key), используемом для шифрования данных.
- Чтобы предотвратить уязвимость пароля в отображаемой версии файла конфигурации или в его резервной копии, хранящейся на сервере, некоторые пароли можно зашифровать, используя глобальную команду конфигурации `service password-encryption`.
- Команда `banner` режима глобальной конфигурации применяется для настройки трех следующих типов сообщений.
  - *Сообщение дня* (MOTD) отображается до того, как появится приглашение аутентификации. Используется для некоторых временных сообщений, например, "Маршрутизатор будет выключен в полночь для технического осмотра".

- *Сообщение перед аутентификацией* отображается до выполнения аутентификации, но после сообщения дня. Обычно используется для какого-либо постоянного сообщения, например: "Неавторизованный доступ к устройству запрещен".
- *Сообщение после аутентификации* отображается после успешной аутентификации пользователя. Обычно используется для вывода информации, которая должна быть скрыта от неавторизованных пользователей.

- Коммутатор настраивает свой IPv4-адрес и маску на специальном, подобном сетевой плате интерфейсе VLAN. Ниже приведена последовательность действий по настройке IPv4-адреса на интерфейсе VLAN 1 коммутатора.

  Этап 1 Перейти в режим конфигурации сети VLAN 1 с помощью команды `interface vlan 1` из глобального режима конфигурации устройства

  Этап 2 Присвоить IP-адрес и маску с помощью команды `ip address` *`ip-адрес маска`* в подрежиме конфигурации интерфейса

  Этап 3 Включить виртуальный интерфейс сети VLAN 1 с помощью команды `no shutdown` в подрежиме конфигурации интерфейса

  Этап 4 Указать стандартный шлюз устройства в глобальном режиме конфигурации с помощью команды `ip default-gateway` *`ip-адрес`*

  Этап 5 Добавить глобальную команду `ip name-server` *`ip-адрес1 ip-адрес2...`*, чтобы настроить коммутатор на использование DNS при поиске имен по их IP-адресам (необязательно)

- Чтобы административно включить интерфейс на коммутаторе, используйте подкоманду интерфейса `no shutdown`, а чтобы отключить его — подкоманду интерфейса `shutdown`.
- Если сетевой инженер точно знает, какие конкретно устройства будут подключены кабелями к каким интерфейсам коммутатора, он может использовать защиту порта, чтобы его могли использовать только указанные устройства. Это затрудняет злоумышленнику возможность подключить портативный компьютер к неиспользуемому порту коммутатора. Когда недозволенное устройство пытается посылать фреймы на интерфейс коммутатора, он может предпринять в ответ различные действия, от вывода информационных сообщений до отключения интерфейса.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Предположим, коммутатору с консоли были отданы команды `enable secret` и `enable password`. Вам необходимо отключиться от коммутатора и опять подключиться к нему через консоль. Пароль какой из заданных команд нужно ввести?

   А) `enable password`.

   Б) `enable secret`.

   В) Пароля не будет.

   Г) Пароль команды `password`, если она была введена.

2. Сетевой инженер настроил коммутатор Cisco 2960 таким образом, что к нему разрешен доступ Telnet с паролем `mypassword`. Позже инженер изменил конфигурацию так, чтобы доступ был возможен по защищенному соединению SSH. Какие из указанных ниже команд вошли в новую конфигурацию? (Выберите два ответа.)

   А) `username` *имя* `password` *пароль* в режиме vty.

   Б) `username` *имя* `password` *пароль* в режиме глобальной конфигурации устройства.

   В) `login local` в режиме vty.

   Г) `transport input ssh` в режиме глобальной конфигурации устройства.

3. Следующая команда была скопирована из буфера в командную строку режима конфигурации коммутатора компании Cisco в сеансе Telnet `banner login this is the login banner`. Что из перечисленного ниже правильно описывает, как будет выглядеть экран при последующем доступе к консоли устройства?

   А) Текст сообщения отображаться не будет.

   Б) Текст сообщения будет состоять из фразы "`his is`".

   В) Текст сообщения будет состоять из фразы "`this is the login banner`".

   Г) Текст сообщения будет состоять из фразы "`Login banner configured. no text defined`" (Сообщения настроены, но текст не задан.)

4. Какое действие не является обязательным при настройке защиты порта без автоматического обнаружения MAC-адресов?

   А) Указать максимально разрешенное количество MAC-адресов для интерфейса с помощью команды `switchport port-security maximum` в режиме конфигурации интерфейса.

   Б) Включить режим безопасности порта с помощью команды конфигурации интерфейса `switchport port-security`.

   В) Указать разрешенные MAC-адреса с помощью команды `switchport port-security mac-address` в режиме конфигурации интерфейса.

   Г) Все указанные выше команды являются обязательными.

5. Персональный компьютер сетевого инженера подключен к коммутатору в главном офисе. К маршрутизатору главного офиса все филиалы подключены через последовательные интерфейсы, в каждом филиале есть малый маршрутизатор и коммутатор. Какие команды и в каком режиме конфигурации должны быть введены в устройства, чтобы инженер мог установить сеансы Telnet с коммутаторами всех филиалов? (Выберите три ответа.)

   А) Ввести команду `ip address` в режиме конфигурации сети VLAN1.

   Б) Ввести команду `ip address` в режиме глобальной конфигурации устройства.

   В) Ввести команду `ip default-gateway` в режиме конфигурации сети VLAN1.

   Г) Ввести команду `ip default-gateway` в режиме глобальной конфигурации устройства.

   Д) Ввести команду `password` в режиме конфигурации консольной линии.

   Е) Ввести команду `password` в режиме конфигурации виртуальных терминалов (vty).

6. Какая из приведенных ниже команд отключает согласование скорости согласно стандарту IEEE для порта 10/100 коммутатора Cisco?

   А) `negotiate disable` в режиме конфигурации интерфейса.

   Б) `no negotiate` в режиме конфигурации интерфейса.

   В) `speed 100` в режиме конфигурации интерфейса.

   Г) `duplex half` в режиме конфигурации интерфейса.

   Д) `duplex full` в режиме конфигурации интерфейса.

   Е) `speed 100` и `duplex full` в режиме конфигурации интерфейса.

7. В каком режиме командной строки (CLI) можно задать настройки дуплексного режима для интерфейса Fast Ethernet 0/5?

   А) В пользовательском.

   Б) В привилегированном.

   В) В режиме глобальной конфигурации.

   Г) В режиме VLAN.

   Д) В режиме конфигурации интерфейса.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 8.4.

Таблица 8.4. Ключевые темы главы 8

Элемент	Описание	Страница
Прим. 8.1	Настройка простых паролей и имен хостов	251
Рис. 8.2	Настройка на коммутаторе аутентификации по локальному имени пользователя и паролю	253
Список	Этапы настройки протокола SSH на коммутаторе	254
Список	Ключевые факты о командах enable secret и enable password	259
Табл. 8.2	Команды буфера истории команд	262
Рис. 8.5	Концепция виртуального интерфейса коммутатора (SVI)	265
Список	Настройка IP-адреса и стандартного шлюза коммутатора	265
Список	Настройка конфигурации коммутатора на изучение параметров IP в режиме клиента DHCP	266
Список	Основные варианты защиты порта	270
Список	Последовательность настройки защиты порта	271
Табл. 8.3	Действия при нарушении защиты порта	274
Список	Рекомендованные настройки неиспользуемых портов коммутатора	275

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

Telnet, SSH, локальное имя пользователя (local username), интерфейс VLAN (VLAN interface), защита порта (port security)

Таблицы команд

В табл. 8.5–8.9 приведены команды конфигурации этой главы с описанием, а в табл. 8.10 — список пользовательских команд.

Таблица 8.5. Команды регистрации Console, Telnet и SSH

Команда	Режим/назначение/описание
`line console 0`	Переключает контекст командной строки в режим конфигурации консольной линии
`line vty` *`1-я_линия_vty последняя_линия_vty`*	Переключает контекст в режим конфигурации линий vty для указанного в команде диапазона
`login`	Режим конфигурации консольной линии или линии vty. Указывает операционной системе IOS, что нужно выдавать приглашение для ввода пароля
`password` *`пароль`*	Режим конфигурации консольной линии или линии vty. Задает пароль, который будет запрашиваться при подключении к устройству
`login local`	Режим конфигурации консольной линии или линии vty. Указывает операционной системе IOS, что нужно выдавать приглашение для ввода имени пользователя и пароля, которые проверяются в локальной базе данных (т.е. заданных с помощью команды `username` в режиме глобальной конфигурации маршрутизатора или коммутатора)
`username` *`имя`* `password` *`пароль`*	Режим глобальной конфигурации. С помощью этой команды можно задавать множество пар "имя пользователя–пароль" для аутентификации. Имена и пароли используются командой `login local` в режиме конфигурации линий
`crypto key generate rsa`	Режим глобальной конфигурации. Создает и сохраняет (в закрытой области флеш-памяти) ключи протокола SSH
`transport input {telnet | ssh}`	Режим конфигурации линии vty. Указывает, разрешен ли к устройству доступ по протоколам Telnet, SSH и др. Если в команде указаны оба протокола, значит, доступ возможен как с помощью сеанса Telnet, так и SSH (стандартно)
`service password-encryption`	Глобальная команда, шифрующая (слабо) пароли, определенные командами `username password`, `enable password` и `login`

Таблица 8.6. Команды конфигурации коммутатора IPv4

Команда	Режим/назначение/описание
`interface vlan` *номер*	Переключает контекст командной строки в режим конфигурации интерфейса сети VLAN. Для сети VLAN 1 устанавливается IP-адрес
`ip address` *адрес маска_подсети*	Режим конфигурации интерфейса. Используется для статического указания IP-адреса и маски подсети
`ip address dhcp`	Режим конфигурации интерфейса. Настраивает коммутатор как клиент DHCP, чтобы он мог автоматически получить IP-адрес, маску подсети и стандартный шлюз
`ip default-gateway` *адрес*	Режим глобальной конфигурации. Задает адрес стандартного шлюза для коммутатора. Эта команда не нужна, если коммутатор использует протокол DHCP
`ip name-server` *сервер-ip-1 сервер-ip-2 ...*	Глобальная команда конфигурации IP-адресов серверов DNS, чтобы любые команды после регистрации на коммутаторе могли использовать DNS для преобразования имен

Таблица 8.7. Команды конфигурации интерфейса коммутатора

Команда	Режим/назначение/описание
`interface` *тип номер_порта*	Переключает контекст командной строки в режим конфигурации интерфейса. В параметре типа интерфейса обычно указывают значение `FastEthernet` или `GigabitEthernet` (для коммутаторов). Возможные номера портов зависят от модели коммутатора и выглядят, например, так: 0/1, 0/2 и т.п.
`interface range` *тип диапазон_портов*	Переключает контекст командной строки в режим конфигурации диапазона интерфейсов. Вводимые после этой команды настройки применяются ко всем интерфейсам диапазона
`shutdown` `no shutdown`	Режим конфигурации интерфейса. Первая команда административно выключает интерфейс, вторая, соответственно, включает
`speed {10 | 100 | 1000 | auto}`	Режим конфигурации интерфейса. Задает скорость интерфейса, если указан параметр `auto`, выполняется автосогласование скорости линии
`duplex {auto | full | half}`	Режим конфигурации интерфейса. Задает дуплексный режим работы интерфейса, если указан параметр `auto`, выполняется автосогласование дуплексности
`description` *текст*	Режим конфигурации интерфейса. Устанавливает текстовый комментарий-описание для интерфейса. Используется для упрощения поиска, устранения неисправностей и удобства в работе

Таблица 8.8. Команды защиты порта

`switchport mode {access | trunk | negotiate}`	Режим конфигурации интерфейса. Указывает коммутатору всегда быть портом доступа, или всегда быть портом магистрального канала, или вести переговоры
`switchport port-security mac-address` *mac-адрес*	Режим конфигурации интерфейса. Добавляет статическую запись для указанного MAC-адреса в таблицу разрешенных для данного интерфейса
`switchport port-security mac-address sticky`	Режим конфигурации интерфейса. Разрешает коммутатору самостоятельно обнаружить MAC-адреса и добавить их в таблицу безопасных для текущего интерфейса
`switchport port-security maximum` *значение*	Режим конфигурации интерфейса. Указывает максимально разрешенное количество безопасных MAC-адресов для данного интерфейса
`switchport port-security violation {protect | restrict | shutdown}`	Режим конфигурации интерфейса. Указывает, какое действие должен предпринять коммутатор в том случае, если нарушен режим безопасности и устройство с небезопасным MAC-адресом пытается получить доступ к сети через защищенный порт

Таблица 8.9. Дополнительные команды конфигурации коммутатора

Команда	Режим/назначение/описание
`hostname` *имя*	Режим глобальной конфигурации. Задает имя хоста для коммутатора, которое также используется в приглашении командной строки
`enable secret` *пароль*	Режим глобальной конфигурации. Задает пароль привилегированного режима
`history size` *число*	Режим конфигурации линии. Задает количество команд, которое будет сохраняться в буфере истории команд
`logging synchronous`	Режим консоли или vty. Заставляет IOS выводить регистрационные сообщения в естественных паузах между командами, а не посередине строки вывода
`[no] logging console`	Режим глобальной конфигурации. Разрешает или запрещает вывод на консоль регистрационных сообщений
`exec-timeout` *минуты [секунды]*	Режим консоли или vty. Устанавливает период бездеятельности так, чтобы по его истечении IOS закрыла текущий пользовательский сеанс
`switchport access vlan` *номер-сети*	Подкоманда интерфейса, определяющая сеть VLAN, в которой располагается интерфейс
`banner [motd | exec | login]` *разделитель текст_банера разделитель*	Режим глобальной конфигурации. Задает баннер, отображаемый при регистрации пользователя на коммутаторе или маршрутизаторе

Таблица 8.10. Список пользовательских команд главы 8

Команда	Назначение
`show running-config`	Выводит текущую конфигурацию
`show running-config | begin line vty`	Выдает текущую конфигурацию устройства, начиная с той строки, в которой встречается текст `line vty`
`show mac address-table dynamic`	Показывает динамические записи в таблице коммутации коммутатора
`show dhcp lease`	Показывает информацию, связанную с режимом клиента DHCP коммутатора. Информация содержит IP-адрес, сетевую маску и адрес стандартного шлюза
`show crypto key mypubkey rsa`	Показывает публичный и общий ключи, которые были созданы для протокола SSH с помощью команды `crypto key generate rsa` режима глобальной конфигурации устройства
`show ip ssh`	Выдает информацию о состоянии сервера SSH, включая версию протокола SSH
`show ssh`	Показывает информацию о пользователях, подключенных к маршрутизатору сеансами протокола SSH
`show interfaces status`	Показывает краткий список и состояние интерфейсов (одна строка на интерфейс). Выводимая командой информация содержит описание, состояние, а также настройки дуплексности и скорости для интерфейса
`show interfaces vlan 1`	Показывает состояние интерфейса, IP-адрес коммутатора, его сетевую маску и еще много полезной информации
`show ip default-gateway`	Показывает состояние интерфейса, IP-адрес коммутатора, его сетевую маску и еще много полезной информации
`show port-security interface` *тип номер*	Показывает параметры конфигурации режима безопасности порта и его состояние
`terminal history size` *x*	Позволяет задавать размер буфера истории команд (*x*) только для текущего сеанса пользователя
`show history`	Отображает команды, находящиеся в буфере истории команд

Ответы на контрольные вопросы:

1 Б. **2** Б и В. **3** Б. **4** Б. **5** А, Г и Е. **6** Е. **7** Д.

Реализация виртуальных локальных сетей

Коммутаторы Ethernet получают фреймы Ethernet, принимают решение, а затем перенаправляют (коммутируют) полученные фреймы. Эта базовая логика основана на MAC-адресах интерфейсов, на которые поступают фреймы и на которые коммутатор перенаправляет их. На решение коммутатора о перенаправлении фреймов оказывают влияние многие факторы, но из всех рассматриваемых в данной книге наибольшее влияние оказывают *виртуальные локальные сети* (VLAN).

Эта глава посвящена концепции VLAN и их настройке. В первом разделе обсуждаются основные концепции, включая влияние сетей VLAN на отдельный коммутатор, использование магистрального соединения для создания сетей VLAN, охватывающих несколько коммутаторов, и перенаправление трафика между сетями VLAN с использованием маршрутизатора. Во втором разделе демонстрируется настройка сетей VLAN и их магистральных каналов, включая статическое присвоение интерфейсов.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Передача данных между двумя хостами по сети.

Технологии коммутации сетей LAN

Базовые концепции коммутации и работа коммутаторов Cisco.

Широковещательные домены.

Таблица CAM.

Создание логических сегментов сети VLAN и необходимость маршрутизации между ними.

Принцип сегментации сети и базовые концепции управления трафиком.

Настройка и проверка сети VLAN.

Настройка и проверка магистрального соединения на коммутаторах Cisco.

Протокол DTP.

Поиск и устранение неисправностей

Поиск неисправностей и решение проблем сетей VLAN.

Идентификация настроенных сетей VLAN.

Исправление принадлежности порта.

Настройка IP-адреса.

Поиск неисправностей и решение проблем магистрального соединения на коммутаторах Cisco.

Исправление состояния магистрального канала.

Исправление конфигурации инкапсуляции.

Исправление разрешенных VLAN.

Основные темы

Концепции виртуальных локальных сетей

Прежде чем приступить к изучению VLAN, сначала имеет смысл выяснить, что это такое. С одной стороны, локальная сеть включает все пользовательские устройства, серверы, коммутаторы, маршрутизаторы, кабели и беспроводные точки доступа в одной области. Но для концепции виртуальной сети LAN больше подходит другое определение локальной сети:

Локальная сеть (LAN) объединяет все устройства в том же широковещательном домене.

Широковещательный домен объединяет все устройства, подключенные к сети LAN, таким образом, что когда любое из устройств посылает широковещательный фрейм, все остальные устройства получают его копию. Таким образом, с другой стороны, локальная сеть и широковещательный домен — это одно и то же.

Без виртуальных сетей коммутатор полагает, что все его интерфейсы находятся в том же широковещательном домене. Таким образом, когда на один порт коммутатора поступает широковещательный фрейм, он перенаправляет его на все остальные порты. Согласно этой логике, чтобы создать два разных широковещательных домена (или LAN), необходимо купить два разных коммутатора Ethernet (рис. 9.1).

Рис. 9.1. Создание двух широковещательных доменов с двумя физическими коммутаторами и без сетей VLAN

При поддержке VLAN тех же целей (создание двух широковещательных доменов) может достичь один коммутатор, как показано на рис. 9.1. Коммутатор VLAN может настроить часть интерфейсов на один широковещательный домен, а часть на другой, создав в результате два широковещательных домена. Эти созданные коммутатором индивидуальные широковещательные домены и являются *виртуальными локальными сетями* (virtual LAN — VLAN).

На рис. 9.2 представлен один коммутатор, создающий две сети VLAN, его порты для каждой из них являются полностью независимыми. Коммутатор никогда не перенаправит фрейм, посланный компьютером Дино (VLAN 1), через порты на компьютер Вилмы или Бетти (VLAN 2).

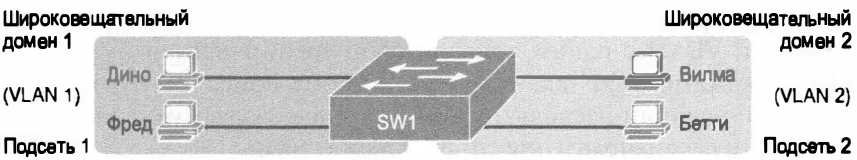


Рис. 9.2. Создание двух широковещательных доменов с использованием одного коммутатора и сети VLAN

Проект территориальной локальной сети, в котором больше сетей VLAN с меньшим количеством устройств в каждой, зачастую имеет больше преимуществ. Например, широковещательное сообщение, посланное одним хостом VLAN, будет получено и обработано всеми другими хостами данной VLAN, но не хостами в других VLAN. Ограничение количества хостов, получающих каждый широковещательный фрейм, снижает количество хостов, впустую тратящих ресурсы на обработку ненужных им широковещательных сообщений. Это снижает также риск нарушения безопасности, поскольку фреймы, посланные любым хостом, поступают на меньшее количество хостов. И это лишь некоторые из причин разделения хостов на отдельные сети VLAN. Ниже приведен список наиболее распространенных причин создания меньших широковещательных доменов (сетей VLAN):

Ключевая тема **Причины применения сетей VLAN**

- Сокращение дополнительных затрат процессоров всех устройств за счет сокращения количества устройств, получающих каждый широковещательный фрейм.
- Улучшение защиты за счет сокращения количества хостов, получающих копии фреймов при их лавинной рассылке коммутатором (широковещание, групповая передача и одноадресатные фреймы с неизвестным получателем).
- Улучшение защиты хостов, пересылающих важные данные, за счет их помещения в отдельную сеть VLAN.
- Возможность более гибкого объединения пользователей в группы (например, по отделам) вместо физического разделения по местоположению.
- Упрощение поиска проблемы в сети, поскольку большинство проблем локализуется в области набора устройств, формирующих широковещательный домен.
- Сокращение дополнительных затрат на работу протокола распределенного связующего дерева (STP) за счет ограничения VLAN одним коммутатором доступа.

Причины применения сетей VLAN подробно не рассматриваются в этой главе, достаточно лишь знать, что они используются в большинстве корпоративных сетей. В оставшейся части данной главы рассматривается механика работы сетей VLAN на нескольких коммутаторах Cisco, включая необходимую настройку. Для этого в следующем разделе исследуется магистральное соединение VLAN — обязательное средство при установке сети VLAN, содержащей несколько коммутаторов LAN.

Создание сети VLAN при нескольких коммутаторах и магистральном соединении

Настройка сети VLAN с одним коммутатором требует немного усилий: достаточно настроить каждый порт так, чтобы указать ему номер VLAN, к которой он принадлежит. При наличии нескольких коммутаторов следует учитывать дополнительные концепции перенаправления трафика между ними.

Когда сети VLAN используются в сетях с несколькими соединенными между собой коммутаторами, на каналах связи между ними применяется *магистральное со-*

единение VLAN (VLAN trunking). Магистральное соединение VLAN подразумевает использование коммутаторами процесса *назначения тегов VLAN* (VLAN tagging), когда передающий коммутатор добавляет к фрейму другой заголовок перед его передачей по магистральному каналу. Этот дополнительный заголовок включает поле *идентификатора VLAN* (VLAN ID), позволяющего передающему коммутатору ассоциировать фрейм с конкретной сетью VLAN, а получающему коммутатору узнать, к какой именно VLAN принадлежит данный фрейм.

На рис. 9.3 приведен пример двух сетей VLAN с несколькими коммутаторами, но без магистрального соединения. Здесь используются две сети VLAN: VLAN 10 и VLAN 20. Каждой сети VLAN присвоено по два порта на каждом коммутаторе, поэтому каждая сеть VLAN существует в обоих коммутаторах. Для перенаправления трафика сети VLAN 10 между двумя коммутаторами проект подразумевает наличие канала связи между ними, который полностью находится в сети VLAN 10. Аналогично для обеспечения трафика сети VLAN 20 между коммутаторами расположен второй канал связи, уже полностью расположенный в сети VLAN 20.

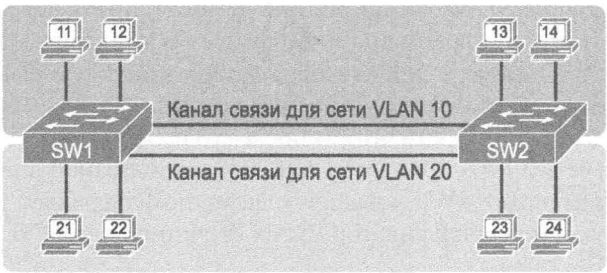


Рис. 9.3. Сети VLAN при наличии нескольких коммутаторов, но без магистрального соединения

Проект, показанный на рис. 9.3, работает прекрасно. Компьютер PC11 (в сети VLAN 10) вполне может послать фрейм компьютеру PC14. Фрейм попадет на коммутатор SW1, а затем по каналу связи (для VLAN 10) на коммутатор SW2. Но хотя этот проект работает, его масштабирование не так просто. Для поддержки каждой сети VLAN требуется отдельный физический канал связи между коммутаторами. Если бы понадобилось 10 или 20 сетей VLAN, то между коммутаторами пришлось бы проложить 10 или 20 каналов связи и использовать для них 10 или 20 портов на каждом коммутаторе.

Концепции назначения тегов VLAN

Магистральное соединение VLAN создает между коммутаторами один канал связи, способный поддерживать столько сетей VLAN, сколько необходимо. Коммутаторы рассматривают магистральный канал как часть всех VLAN. Тем не менее трафик в магистральном канале VLAN остается раздельным, и фреймы VLAN 10 никоим образом не попадут на устройства VLAN 20 (и наоборот), поскольку, пересекая магистральный канал, каждый фрейм идентифицирован номером VLAN. На рис. 9.4 приведена концепция сети с одним физическим каналом связи между двумя коммутаторами.

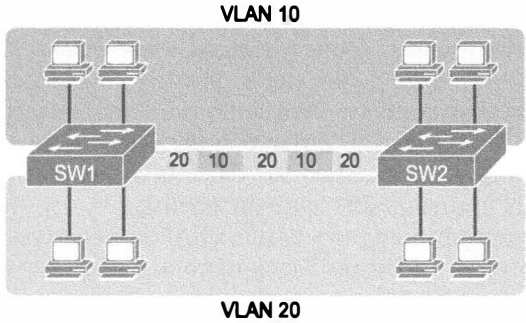


Рис. 9.4. Сети VLAN с несколькими коммутаторами и магистральным соединением

Магистральное соединение позволяет коммутаторам передавать фреймы нескольких сетей VLAN по одному физическому каналу за счет добавления небольшого заголовка к фрейму Ethernet. Пример на рис. 9.5 демонстрирует передачу компьютером PC11 широковещательного фрейма на интерфейсе Fa0/1 (этап 1). Для лавинной рассылки коммутатор SW1 должен перенаправить широковещательный фрейм на коммутатор SW2. Но коммутатор SW1 должен как-то дать знать коммутатору SW2, что фрейм принадлежит сети VLAN 10, чтобы после его получения осуществить лавинную рассылку только в сети VLAN 10, а не VLAN 20. Как показано на этапе 2, перед передачей фрейма коммутатор SW1 добавил к исходному фрейму Ethernet заголовок VLAN, в котором указан идентификатор VLAN (в данном случае VLAN 10).

Ключевая тема

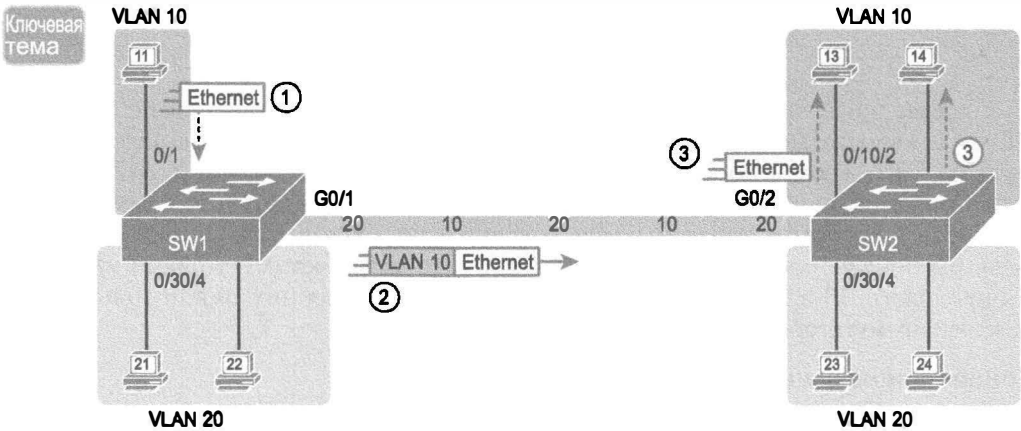


Рис. 9.5. Магистральное соединение VLAN между двумя коммутаторами

Когда коммутатор SW2 получает фрейм, он понимает, что фрейм принадлежит сети VLAN 10. Затем коммутатор SW2 удаляет заголовок VLAN и перенаправляет первоначальный фрейм по интерфейсу к VLAN 10 (этап 3).

Для другого примера рассмотрим случай, когда компьютер PC21 (в сети VLAN 20) посылает широковещательный фрейм. Коммутатор SW1 посылает его через порт Fa0/4 (поскольку этот порт находится в сети VLAN 20) и порт Gi0/1 (поскольку это магистральный канал, а значит, он поддерживает несколько разных сетей VLAN).

Коммутатор SW1 добавляет к фрейму заголовок магистрали, содержащий идентификатор VLAN 20. Выяснив, что фрейм принадлежит сети VLAN 20, коммутатор SW2 удалит магистральный заголовок и перенаправит его только на порты Fa0/3 и Fa0/4, поскольку они находятся в сети VLAN 20, но не на порты Fa0/1 и Fa0/2, так как они находятся в сети VLAN 10.

Протоколы магистралей VLAN 802.1Q и ISL

В последние годы компания Cisco использует два протокола магистральных соединений: *протокол межкоммутаторных соединений* (Inter-Switch Link — ISL) и протокол 802.1Q стандарта IEEE. Компания Cisco использовала протокол ISL задолго до появления протокола 802.1Q частично потому, что IEEE еще не определил стандарт для магистралей VLAN. Несколько лет назад IEEE закончил работу над стандартом 802.1Q, определяющим иной способ создания магистральных соединений. Сейчас протокол 802.1Q стал наиболее популярным протоколом магистральных соединений, и компания Cisco больше не поддерживает стандарт ISL на некоторых более новых моделях коммутаторов LAN, включая 2960, который используется в примерах этой книги.

Хотя оба протокола отмечают каждый фрейм идентификатором VLAN, детали процесса у них разные. Протокол 802.1Q использует дополнительное 4-байтовое поле — заголовок 802.1Q в заголовке Ethernet первоначального фрейма, как показано на рис. 9.6, *сверху*. Что касается полей в заголовке 802.1Q, то поле идентификатора VLAN занимает только 12 битов, но для тем данной книги это не имеет значения. Теоретически это 12-битовое поле способно идентифицировать максимум 2^{12} (4096) сетей VLAN, хотя на практике доступно максимум 4094 значения. (Согласно стандартам 802.1Q и ISL, поле идентификатора VLAN имеет два зарезервированных значения — 0 и 4095.)

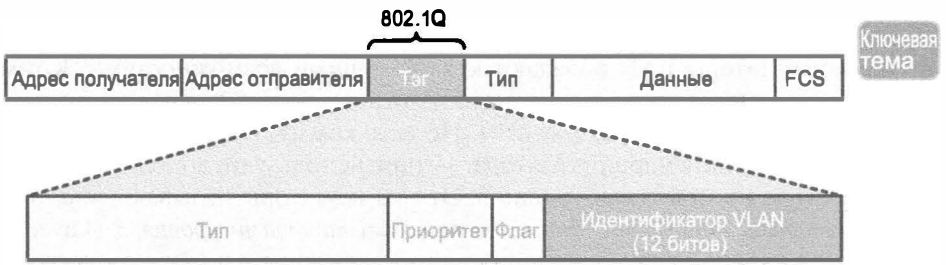


Рис. 9.6. Заголовок магистрального соединения по стандарту 802.1Q

Коммутаторы Cisco разделяют диапазон идентификаторов VLAN (1-4094) на два диапазона: нормальный и расширенный. Все коммутаторы могут использовать нормальный диапазон идентификаторов VLAN со значениями 1–1005, и только некоторые могут использовать расширенный диапазон от 1005 до 4094. Правила использования коммутаторами расширенного диапазона идентификаторов VLAN зависят от конфигурации *протокола создания магистралей VLAN* (VLAN Trunking Protocol — VTP), кратко обсуждаемого ниже.

Для каждого магистрального канала стандарт 802.1Q определяет также один специальный идентификатор VLAN, обозначающий *собственную сеть VLAN* (native

VLAN) (стандартно это VLAN 1). По определению протокол 802.1Q не добавляет заголовок 802.1Q к фреймам в собственной сети VLAN. Когда коммутатор с другой стороны магистрального канала получает фрейм без заголовка 802.1Q, он понимает, что фрейм принадлежит собственной сети VLAN. Из-за этого оба коммутатора должны "договориться", какую сеть VLAN считать собственной.

Согласно стандарту 802.1Q, собственная сеть VLAN обладает некими уникальными функциями: она, например, способна обеспечивать соединения с устройствами, которые не поддерживают магистральное соединение. Например, коммутатор Cisco может быть подключен к коммутатору, который не поддерживает магистральные соединения 802.1Q. Коммутатор Cisco мог бы послать фреймы со значением собственной сети VLAN, т.е. без магистрального заголовка, и другой коммутатор будет понимать их. Концепция собственной сети VLAN позволяет коммутаторам передавать трафик как минимум одной сети VLAN (собственной VLAN), поддерживая некоторые базовые функции, такие как доступность коммутатора по Telnet.

Перенаправление данных между сетями VLAN

При создании территориальной локальной сети, содержащей много сетей VLAN, требуется обеспечить всем устройствам возможность передавать данные на все остальные устройства. Давайте обсудим некоторые из концепций перенаправления данных между сетями VLAN.

В первую очередь это поможет усвоить терминологию коммутаторов LAN. Все функции и логика коммутаторов Ethernet, описанные до сих пор, соответствовали протоколам уровня 2 модели OSI. Например, в главе 6 упоминалось о том, что коммутаторы LAN получают фреймы Ethernet (концепция уровня 2), распознают MAC-адрес получателя (адрес уровня 2) и перенаправляют фрейм Ethernet на другой интерфейс. В этой главе уже упоминалась концепция сетей VLAN как широковещательных доменов, что тоже является концепцией уровня 2.

Хотя некоторые коммутаторы LAN работают так, как описывалось до сих пор, другие коммутаторы LAN обладают куда большими возможностями. Коммутаторы LAN, передающие данные на основании логики уровня 2, зачастую называют *коммутаторами уровня 2* (Layer 2 switch). Но есть коммутаторы, способные выполнять некоторые функции маршрутизатора, — они используют дополнительную логику, определенную протоколами уровня 3. Эти коммутаторы называют *многоуровневыми коммутаторами* (multilayer switch), или *коммутаторами уровня 3* (Layer 3 switch). Этот раздел начинается с обсуждения перенаправления данных между сетями VLAN коммутаторами уровня 2, а завершается обсуждением применения для этого коммутаторов уровня 3.

Маршрутизация пакетов между сетями VLAN с использованием маршрутизатора

При включении виртуальной локальной сети (VLAN) в проект территориальной локальной сети все устройства сети VLAN должны быть в той же подсети. Согласно той же логике, устройства в разных сетях VLAN должны принадлежать разным подсетям. Например, два компьютера, показанные на рис. 9.7, *слева*, находятся в сети VLAN 10 и в подсети 10. Два компьютера, показанные справа, находятся в другой сети VLAN (20) и в другой подсети (20).

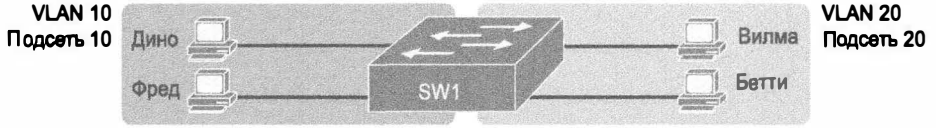


Рис. 9.7. Маршрутизация на коммутаторах между двумя физически отделенными сетями VLAN

ВНИМАНИЕ!

Подсети на рисунке обозначены несколько абстрактно, как "подсеть 10", чтобы не отвлекаться на номера подсетей. Обратите также внимание на то, что номера подсетей не должны совпадать с номерами сетей VLAN.

Коммутаторы уровня 2 не будут перенаправлять данные между двумя сетями VLAN. Фактически одна из задач сетей VLAN заключается в том, чтобы отделить трафик одной виртуальной сети от другой и предотвратить попадание фреймов из одной сети VLAN в другую. Например, если компьютер Дино (VLAN 10) пошлет любой фрейм Ethernet на коммутатор SW1 уровня 2, то коммутатор не станет перенаправлять его на компьютеры справа, находящиеся в сети VLAN 20.

Сеть в целом должна обеспечивать передачу трафика, входящего и исходящего из каждой сети VLAN, даже при том, что коммутатор уровня 2 не перенаправляет фреймы за пределы виртуальной сети. Задачу перенаправления данных между сетями VLAN выполняет маршрутизатор. Вместо коммутации фреймов Ethernet уровня 2 между двумя сетями VLAN сеть должна перенаправлять между этими двумя подсетями пакеты уровня 3.

Поскольку в предыдущем абзаце встретилась довольно специфическая формулировка, связанная с уровнями 2 и 3, уделим минуту этой теме. Логика уровня 2 не позволяет коммутатору уровня 2 перенаправлять фреймы Ethernet уровня 2 (L2PDU) между сетями VLAN. Однако маршрутизаторы могут перенаправить пакеты уровня 3 (L3PDU) между подсетями, как и положено.

На рис. 9.8, например, представлен маршрутизатор, способный перенаправлять пакеты между подсетями 10 и 20. На рисунке демонстрируется тот же коммутатор уровня 2, что и на рис. 9.7, с теми же компьютерами и теми же сетями VLAN и подсетями. Но теперь коммутатор подключен к маршрутизатору R1 одним физическим интерфейсом, принадлежащим сети VLAN 10, и вторым, принадлежащим сети VLAN 20. При соединении с каждой подсетью коммутатор уровня 2 вполне может продолжить перенаправлять фреймы в каждой сети VLAN, в то время как маршрутизатор будет решать задачу о направлении пакетов IP между подсетями.

На рис. 9.8 показан пакет IP, передаваемый компьютером Фреда из одной сети VLAN (подсети) на компьютер Бетти, находящийся в другой сети VLAN (подсети). Коммутатор уровня 2 передает два разных фрейма Ethernet уровня 2: один в сети VLAN 10, от компьютера Фреда на интерфейс F0/0 маршрутизатора R1, и другой, в сети VLAN 20, от интерфейса F0/1 маршрутизатора R1 на компьютер Бетти. С точки зрения уровня 3 компьютер Фреда посылает пакет IP на свой стандартный маршрутизатор (R1), он перенаправляет пакет на другой интерфейс (F0/1) в другую подсеть, где располагается компьютер Бетти.

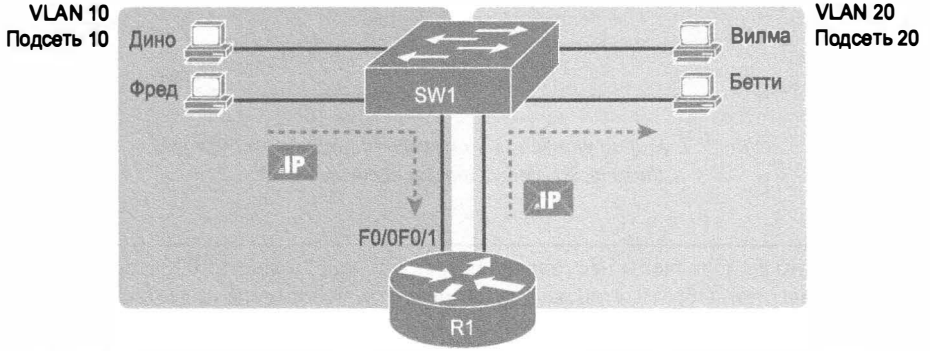


Рис. 9.8. Маршрутизация между двумя сетями VLAN на двух физических интерфейсах

Хотя проект на рис. 9.8 вполне работоспособен, он использует слишком много физических интерфейсов, по одному на каждую VLAN. Намного менее расточительное (и более предпочтительное) решение подразумевает использование магистрального канала VLAN между коммутатором и маршрутизатором. Так, для поддержки всех сетей VLAN достаточно только одного физического канала связи между маршрутизатором и коммутатором. Магистральное соединение возможно между любыми двумя устройствами, способными поддерживать его: между двумя коммутаторами, между маршрутизатором и коммутатором и даже между аппаратными средствами сервера и коммутатором.

На рис. 9.9 представлен концептуально тот же проект, что и на рис. 9.8, с тем же пакетом, следующим от компьютера Фреда к компьютеру Бетти, но теперь маршрутизатор R1 использует магистральное соединение VLAN вместо отдельного канала связи для каждой сети VLAN.

Ключевая тема

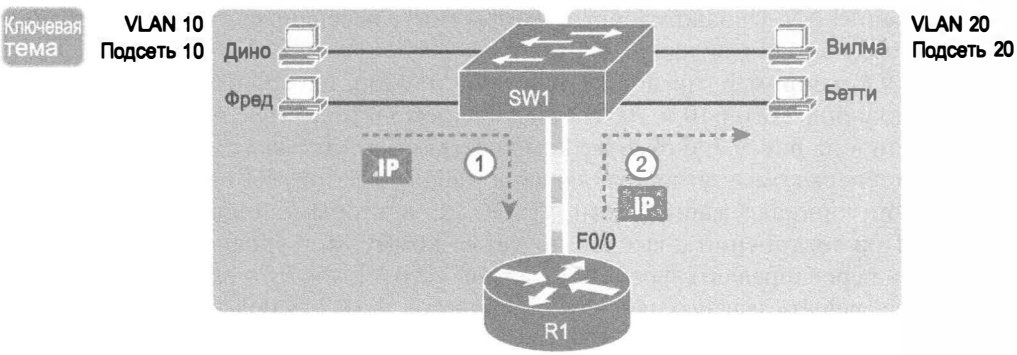


Рис. 9.9. Маршрутизация между двумя сетями VLAN с использованием магистрального канала на маршрутизаторе

ВНИМАНИЕ!

Поскольку у маршрутизатора один физический канал связи, подключенный к коммутатору LAN, такой проект сети иногда вызывают "маршрутизатор на палочке" (router-on-a-stick).

Еще немного терминологии: концепцию на рис. 9.8 и 9.9 иногда упоминают как "маршрутизацию пакетов между сетями VLAN" (routing packets between VLAN). Эту

фразу вполне можно использовать, люди поймут, что имеется в виду. Но для подготовки к экзамену буквально эта фраза неверна, поскольку объединяет маршрутизацию пакетов (концепция уровня 3) и VLAN (концепция уровня 2). Просто "маршрутизация между сетями VLAN" короче, хотя буквально правильно "маршрутизация пакетов уровня 3 между подсетями уровня 3, при сопоставлении каждой из подсетей с различными сетями VLAN уровня 2".

Маршрутизация пакетов коммутаторами уровня 3

У маршрутизации пакетов с использованием физического маршрутизатора (даже при магистральном канале VLAN, как на рис. 9.9) все еще остается одна серьезная проблема: производительность. Физический канал связи налагает ограничение на количество передаваемых битов, а недорогие маршрутизаторы обычно не отличаются высокой мощностью и не могут перенаправлять достаточно много *пакетов за секунду* (packets per second — pps), чтобы поддерживать необходимый объем трафика.

Окончательное решение подразумевает передачу функций маршрутизации аппаратным средствам коммутатора LAN. Производители уже довольно давно начали объединять аппаратные и программные средства коммутаторов уровня 2 с маршрутизаторами уровня 3, выпуская *коммутаторы уровня 3* (они же *многоуровневые коммутаторы*). Коммутаторы уровня 3 могут быть настроены так, чтобы действовать или как только коммутаторы уровня 2, или как коммутаторы уровня 2 с маршрутизаторами уровня 3.

В настоящее время многие средние и крупные корпоративные территориальные локальные сети используют для перенаправления пакетов между подсетями (VLAN) коммутаторы уровня 3.

Концептуально коммутатор уровня 3 работает подобно первоначальным двум устройствам, на базе которых он создан: коммутатора LAN уровня 2 и маршрутизатора уровня 3. Фактически, если понятны концепции перенаправления пакетов на рис. 9.8, при отдельном коммутаторе уровня 2 и маршрутизаторе уровня 3, вы имеете общее представление о работе коммутатора уровня 3, объединяющем все эти функции в одном устройстве. Эта концепция представлена на рис. 9.10, она повторяет многие подробности рис. 9.8, но с дополнительным прямоугольником, демонстрирующим, что один коммутатор уровня 3 выполняет функции коммутатора уровня 2 и маршрутизатора уровня 3.

Эта глава знакомит с основами маршрутизации пакетов IP между сетями VLAN (точнее, между подсетями в сетях VLAN). Конфигурация сетей, использующих внешний маршрутизатор, рассматривается в главе 16, а пока рассмотрим конфигурацию и проверку сетей VLAN и магистральных каналов VLAN.

Конфигурация сетей и магистралей VLAN

Для работы коммутаторам Cisco никакой настройки не требуется. Вполне можно купить коммутатор Cisco, подключить его к соответствующим кабелям, включить, и он заработает. Вам никогда не придется настраивать коммутатор, и он будет прекрасно работать (даже если придется соединять коммутаторы), пока не понадобится несколько сетей VLAN. Но если необходимо использовать несколько сетей VLAN (как в большинстве корпоративных сетей), то некоторая настройка понадобится.

Ключевая тема

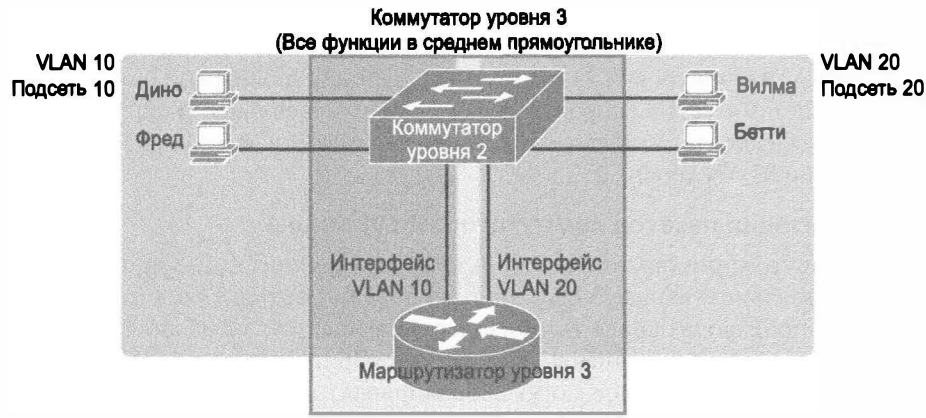


Рис. 9.10. Маршрутизация между сетями VLAN с использованием коммутатора уровня 3

Подробнее конфигурация VLAN рассматривается в двух разделах этой главы. В первом описана настройка интерфейсов доступа, т.е. интерфейсов коммутатора, не используемых для магистральных соединений VLAN. Настройка интерфейсов, используемых для магистральных соединений VLAN, рассматриваются во втором разделе.

Создание сетей VLAN и назначение интерфейсов доступа

В этом разделе показано, как создать сеть VLAN, присвоить ей имя и добавить в нее интерфейсы. Чтобы можно было сосредоточиться на рассмотрении этой темы, в примерах настоящего раздела используется один коммутатор, который не нуждаются в магистрали VLAN.

Чтобы коммутатор Cisco начал перенаправлять фреймы в определенную сеть VLAN, его нужно настроить, указав на существование еще одной сети VLAN. Кроме того, у коммутатора должны быть немагистральные интерфейсы (*интерфейсы доступа* (access interface)), принадлежащие этой сети VLAN, и (или) магистральные каналы, поддерживающие эту VLAN. Этапы настройки интерфейсов доступа приведены ниже, настройка магистрального канала — в соответствующем разделе далее.

Ключевая тема

Последовательность настройки конфигурации VLAN и назначения интерфейсов

Этап 1 Чтобы настроить конфигурацию новой сети VLAN, выполните следующие действия:

A. В режиме настройки конфигурации введите глобальную команду конфигурации `vlan` *`идентификатор_vlan`* для создания сети VLAN и перейдите в режим настройки конфигурации сети VLAN.

B. (Необязательно.) Чтобы присвоить сети VLAN имя, введите подкоманду VLAN `name` *`имя`*. Если этого не сделать, именем VLAN будет `VLAN`*`ZZZZ`*, где *`ZZZZ`* — десятичный идентификатор из четырех цифр

Этап 2 Для каждого интерфейса доступа (интерфейса, принадлежащего не магистральному каналу, а отдельной сети VLAN) выполните следующие действия:

A. Используя команду `interface`, перейдите в режим конфигурации каждого настраиваемого интерфейса.

B. Используя подкоманду интерфейса `switchport access vlan` *`идентификатор_vlan`*, укажите номер сети VLAN, связанной с данным интерфейсом.

C. (Необязательно). Чтобы отключить магистральное соединение на том же интерфейсе и запретить переговоры о создании магистрального канала, используйте подкоманду интерфейса `switchport mode access`.

Хоть этот список и выглядит устрашающе, на самом деле процесс настройки одиночного коммутатора довольно прост. Например, если порты коммутатора следует распределить по трем сетям VLAN (11, 12 и 13), достаточно ввести три команды: `vlan 11`, `vlan 12` и `vlan 13`. Затем для каждого интерфейса введите команду `switchport access vlan 11` (или `12`, или `13`), чтобы присвоить соответствующий интерфейс надлежащей сети VLAN.

Первый пример: полная настройка сети VLAN

В примере 9.1 показан процесс настройки, сводящийся к добавлению новой сети VLAN и назначению интерфейсов доступа к ней. На рис. 9.11 представлена сеть, рассматриваемая в данном примере, с одним коммутатором LAN (SW1) и тремя сетями VLAN (1, 2 и 3), в каждой из которых имеются по два хоста. В этом примере приведены подробные сведения о выполнении двухэтапного процесса настройки сети VLAN 2 и назначения ей интерфейсов; настройка конфигурации сети VLAN 3 рассматривается в следующем примере.

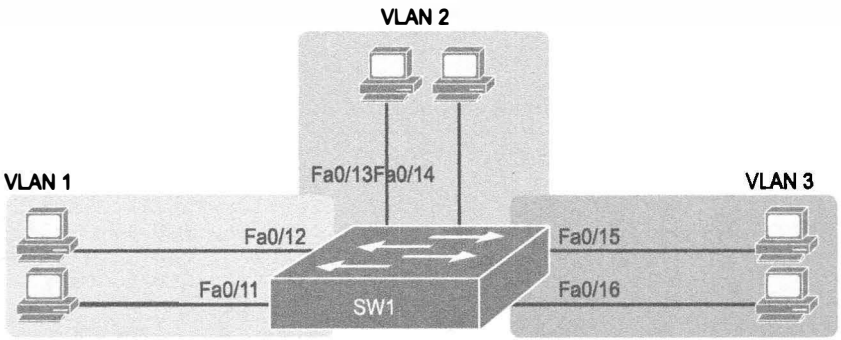


Рис. 9.11. Сеть с одним коммутатором и тремя сетями VLAN

Пример 9.1. Настройка сетей VLAN и назначение им интерфейсов

```
SW1# show vlan brief
VLAN Name                 Status      Ports
---- -------------------- ----------- --------------------------------
1    default              active      Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                      Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                      Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                      Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                      Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                      Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                      Gi0/1, Gi0/2
```

```
1005 trnet-default       act/unsup
! Выше, сети VLAN 2 и 3 еще не существуют. Ниже добавляется сеть VLAN 2,
! командой name Freds-vlan, с присвоением ей двух интерфейсов.

SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# vlan 2
SW1(config-vlan)# name Freds-vlan
SW1(config-vlan)# exit
SW1(config)# interface range fastethernet 0/13 - 14
SW1(config-if)# switchport access vlan 2
SW1(config-if)# end

! Ниже подкоманда интерфейса show running-config выводит списки команд
! на интерфейсах Fa0/13 и Fa0/14.
SW1# show running-config
! Часть строк опущена для краткости
vlan 2
name Freds-vlan
!
! еще часть строк опущена для краткости
interface FastEthernet0/13
  switchport access vlan 2
  switchport mode access
!
interface FastEthernet0/14
  switchport access vlan 2
  switchport mode access
!

SW1# show vlan brief

VLAN Name                 Status     Ports
---- -------------------- ---------- -------------------------------
1 default                 active     Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                     Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                     Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                     Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                     Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                     Fa0/23, Fa0/24, Gi0/1, Gi0/2
2 Freds-vlan              active     Fa0/13, Fa0/14
1002 fddi-default         act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup

SW1# show vlan id 2
VLAN Name                 Status     Ports
---- -------------------- ---------- -------------------------------
2    Freds-vlan           active     Fa0/13, Fa0/14

VLAN Type  SAID   MTU  Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
---- ----- ------ ---- ------ ------ -------- --- -------- ------ -------
2    enet  100010 1500 -      -      -        -   -        0      0

Remote SPAN VLAN
----------------
Disabled
Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------
```

Этот пример начинается с выполнения команды `show vlan brief`, позволяющей убедиться в том, что пять не удаляемых сетей VLAN имеют стандартные параметры, а все интерфейсы назначены сети VLAN 1. (Сеть VLAN 1 не может быть удалена, но может использоваться. На настоящий момент сети VLAN 1002–1005 не могут быть удалены и не могут использоваться для доступа.) Следует отметить, что рассматриваемый коммутатор 2960 имеет 24 порта Fast Ethernet (Fa0/1–Fa0/24) и два порта Gigabit Ethernet (Gi0/1 и Gi0/2), в выводе первой команды все они принадлежали сети VLAN 1.

Далее показан процесс создания сети VLAN 2 и назначение ей интерфейсов Fa0/13 и Fa0/14. Обратите внимание, что в данном примере используется команда `interface range`, применяющая подкоманды интерфейса `switchport access vlan 2` к обоим интерфейсам в диапазоне, что подтверждается выводом команды `show running-config` в конце примера.

После добавления данной конфигурации для получения информации о новой сети VLAN в этом примере повторно выполняется команда `show vlan brief`. В ее выводе указаны сеть VLAN 2, имя `Freds-vlan` и присвоенные ей интерфейсы Fa0/13 и Fa0/14.

Сеть в примере на рис. 9.11 использует в качестве портов доступа шесть интерфейсов коммутатора. Такие порты не должны использовать магистральное соединение, они должны быть присвоены конкретной сети VLAN командой `switchport access vlan` *идентификатор_vlan*. Однако, согласно конфигурации в примере 9.1, эти интерфейсы могли вести переговоры, чтобы стать портами магистрального канала (стандартное состояние коммутатора). Это позволяет порту договориться о магистральном соединении и решить, действовать ли как интерфейс доступа или как магистральный интерфейс.

Для портов, которые всегда должны быть портами доступа, имеет смысл ввести необязательную подкоманду интерфейса `switchport mode access`. Она указывает коммутатору, что порту позволено быть только интерфейсом доступа. Более подробная информация о командах, позволяющих порту вести переговоры об использовании магистральных соединений, приведена в следующем разделе.

Второй пример: сокращенная настройка сети VLAN

Пример 9.1 демонстрирует несколько необязательных команд конфигурации, побочным эффектом которых является немного более продолжительная настройка. Альтернативная конфигурация в примере 9.2 намного короче, в нем добавляется сеть VLAN 3 (как показано на рис. 9.11), ее следует считать продолжением примера 9.1. Обратите также внимание, что до начала этого примера на коммутаторе SW1 отсутствуют данные о сети VLAN 3.

Пример 9.2. Более короткий пример настройки сети VLAN (сети VLAN 3)

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface range Fastethernet 0/15 - 16
SW1(config-if-range)# switchport access vlan 3
% Access VLAN does not exist. Creating vlan 3
SW1(config-if-range)# ^Z
```

```
SW1# show vlan brief

VLAN Name                 Status     Ports
---- -------------------- ---------- -------------------------------
1    default              active     Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                     Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                     Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                     Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                     Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                     Gi0/1, Gi0/2
2    Freds-vlan           active     Fa0/13, Fa0/14
3    VLAN0003             active     Fa0/15, Fa0/16
1002 fddi-default         act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup
```

Пример 9.2 демонстрирует, как коммутатор может динамически создать сеть VLAN (эквивалент глобальной команды конфигурации `vlan` *`идентификатор_vlan`*), когда подкоманда интерфейса `switchport access vlan` применяется к ненастроенной в настоящий момент сети VLAN. В начале этого примера на коммутаторе SW1 отсутствует информация о сети VLAN 3, а когда используется подкоманда интерфейса `switchport access vlan 3`, коммутатор понимает, что такой сети не существует и, как упомянуто в затененном сообщении этого примера, создает сеть VLAN 3, используя стандартное имя (VLAN0003). Никаких других действий по созданию этой сети VLAN не требуются. В конце рассматриваемого процесса обнаруживается, что на коммутаторе существует сеть VLAN 3, а в сети VLAN 3 находятся интерфейсы Fa0/15 и Fa0/16, о чем свидетельствует затененная часть вывода команды `show vlan brief`.

Протокол создания магистралей VLAN (VTP)

Прежде чем перейти к следующим примерам конфигурации, следует узнать о прежнем протоколе Cisco — *протоколе создания магистралей VLAN* (VLAN Trunking Protocol — VTP). Это собственный протокол компании Cisco, выполняющийся на ее коммутаторах. Он анонсирует каждую сеть VLAN, настроенную на одном коммутаторе командой `vlan` *`номер`*, чтобы о ней узнали все остальные коммутаторы в территориальной локальной сети. Однако по разным причинам в большинстве корпоративный сетей протокол VTP не используют.

Эта книга не пропагандирует протокол VTP. Но он оказывает некоторое влияние на работу коммутаторов Cisco Catalyst, даже если и не используется. В этом кратком разделе протокол VTP обсуждается достаточно подробно, чтобы можно было заметить небольшие отличия в его работе.

Каждый коммутатор может работать в одном из трех режимов VTP: серверном, клиентском или прозрачном. Коммутаторы используют серверный и клиентский режимы, когда протокол VTP применяется по прямому назначению: для динамического анонсирования информации о конфигурации сетей VLAN. Однако при множестве коммутаторов Cisco и версий операционной системы IOS протокол VTP не может быть отключен на коммутаторе Cisco полностью; вместо этого коммутатор переводит его в прозрачный режим.

В данной книге мы пытаемся, по возможности, игнорировать протокол VTP. Для этого во всех приведенных примерах коммутаторы используются в прозрачном режиме протокола VTP (глобальная команда `vtp mode transparent`) или при его отключении (глобальная команда `vtp mode off`). Обе позволяют администратору задать как стандартный, так и расширенный диапазон сетей VLAN, а коммутатор перечисляет команды `vlan` в файле конфигурации `running-config`.

И наконец, встретив в практическом применении (выполняя упражнения лабораторных работ с реальными коммутаторами или их эмуляторами) необычное поведение сетей VLAN, проверьте состояние протокола VTP командой `show vtp status`. Если коммутатор будет использовать серверный или клиентский режим VTP, то обнаружится следующее:

- серверные коммутаторы могут настраивать сети VLAN только в стандартном диапазоне (1–1005);
- клиентские коммутаторы не могут настраивать сети VLAN;
- команда `show running-config` не отображает команды `vlan`.

Выполняя практические задания экзаменов CCENT и CCNA по настройке коммутатора, по возможности переведите коммутатор в прозрачный режим или вообще отключите протокол VTP.

ВНИМАНИЕ!

Экспериментируя с параметрами VTP на реальном коммутаторе, будьте очень осторожны. Если этот коммутатор подключен к другим коммутаторам, которые в свою очередь подключены к коммутаторам, используемым в рабочей сети LAN, вполне можно вызвать проблемы, переписав конфигурации VLAN других коммутаторов. Будьте внимательны и никогда не экспериментируйте с параметрами VTP на коммутаторе, если к нему подключены другие коммутаторы, особенно если есть физические каналы связи с рабочими сетями LAN.

Конфигурация магистрального соединения VLAN

Настройка магистрального соединения между двумя коммутаторами Cisco может быть очень простой, если она осуществляется только статически. Например, если два коммутатора Cisco 2960 соединены друг с другом, они поддерживают только протокол 802.1Q, но не ISL. Достаточно добавить буквально одну подкоманду интерфейса для порта коммутатора на каждой стороне канала связи (`switchport mode trunk`), и будет получен магистральный канал VLAN, поддерживаемый всеми сетями VLAN, известными каждому коммутатору.

Однако конфигурация магистрали на коммутаторах Cisco имеет еще много возможностей, включая несколько вариантов для динамических переговоров о разных параметрах магистрали. Они могут быть либо заданы предварительно, либо коммутаторы могут сами договориться о них следующим образом.

- *Тип магистрального соединения*: протокол IEEE, протокол 802.1Q или переговоры о применяемом протоколе.
- *Административный режим*: всегда магистральный канал, никогда магистральный канал или переговоры.

Сначала рассмотрим тип магистрального соединения. Коммутаторы Cisco, поддерживающие протоколы ISL и 802.1Q, способны вести переговоры об используемом типе при помощи *протокола динамического согласования магистральных каналов* (Dynamic Trunk Protocol — DTP). Если оба коммутатора поддерживают оба протокола, они используют протокол ISL; в противном случае они используют общий протокол. Современные коммутаторы Cisco не поддерживают устаревший протокол ISL. Коммутаторы, поддерживающие оба типа магистрального соединения, используют подкоманду интерфейса `switchport trunk encapsulation {dot1q | isl | negotiate}` для того, чтобы задать тип или позволить протоколу DTP договариваться о типе.

Протокол DTP позволяет также согласовать административный режим локальных портов коммутаторов. Под *административным режимом* (administrative mode) подразумевается настройка конфигурации, определяющая, должно ли использоваться магистральное соединение в интерфейсе. У каждого интерфейса также есть *рабочий режим* (operational mode), когда интерфейс выполняет присущие ему действия, возможно, выбранные протоколом DTP в ходе переговоров с другим устройством. Для определения административного режима магистрали коммутаторы Cisco используют подкоманду интерфейса `switchport mode` с параметрами, перечисленными в табл. 9.1.

Ключевая тема

Таблица 9.1. Параметры команды `switchport mode`, определяющие административный режим магистрали

Параметр	Описание
`access`	Всегда быть портом доступа (а не магистрального канала)
`trunk`	Всегда быть портом магистрального канала
`dynamic desirable`	Передавать и отвечать на сообщения переговоров, чтобы динамически решить, использовать ли магистральное соединение
`dynamic auto`	Пассивно ожидать получения сообщений переговоров. При получении таковых вести переговоры об использовании магистрального соединения

В качестве примера рассмотрим два коммутатора на рис. 9.12. Это сеть, показанная на рис. 9.11, дополненная магистральным каналом к новому коммутатору SW2, часть портов которого подключена к сетям VLAN 1 и VLAN 3. Для магистрального соединения оба коммутатора используют канал связи Gigabit Ethernet. В данном случае магистральное соединение не создается динамически, поскольку у обоих коммутаторов (2960) стандартно задан административный режим `dynamic auto`, а это значит, что ни один из них не инициализирует процесс согласования магистрального соединения. После изменения конфигурации одного коммутатора для использования режима `dynamic desirable`, предназначенного для инициализации переговоров, на коммутаторах проводится согласование магистрального соединения, а именно — соединения по протоколу 802.1Q, поскольку коммутаторы 2960 поддерживают только протокол 802.1Q.

Пример 9.3 начинается с отображения стандартной конфигурации двух коммутаторов на рис. 9.12 и позволяет убедиться в отсутствии магистрального соединения между ними.

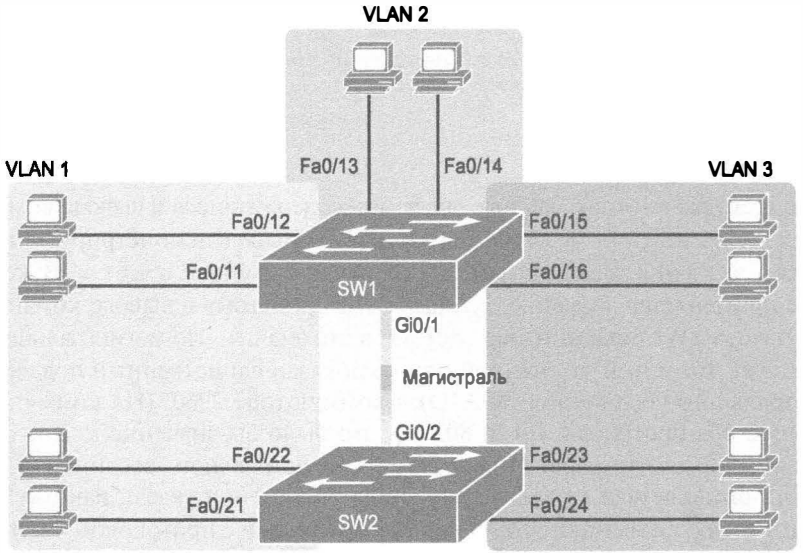


Рис. 9.12. Сеть с двумя коммутаторами и тремя сетями VLAN

Пример 9.3. Изначальное (стандартное) состояние: между коммутаторами SW1 и SW2 нет магистрального соединения

```
SW1# show interfaces gigabit 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dotlq
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dotlq
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
```

```
Appliance trust: none

! Обратите внимание, выполнение следующей команды приводит к
! появлению одной пустой строки вывода.
SW1# show interfaces trunk

SW1#
```

Прежде всего рассмотрим важные сведения, содержащиеся в выводе команды `show interfaces switchport` в начале примера 9.3. Вывод демонстрирует применение стандартного значения административного режима `dynamic auto`. В коммутаторе SW2 также применяется значение `dynamic auto`, поэтому в выводе команды состояния коммутатора SW1 показано как `access`, а это значит, что магистральное соединение отсутствует. В третьей затененной строке показан единственный поддерживаемый тип магистрального соединения (802.1Q) в коммутаторе 2960. (На коммутаторе, поддерживающем оба протокола, ISL и 802.1Q, это было бы значение `negotiate`, означающее переговоры о типе или инкапсуляции магистрального соединения.) Наконец, фактически применяемый тип магистрального соединения указан как `native`, а значит, используется собственная сеть VLAN в соответствии с протоколом 802.1Q.

Пример завершается командой `show interfaces trunk`, но без вывода. Эта команда выводит информацию обо всех интерфейсах магистральных каналов, работающих в настоящий момент, т.е. она перечисляет интерфейсы, которые в настоящее время используют магистральное соединение VLAN. Не перечисляя интерфейсы, эта команда также подтверждает, что канал связи между коммутаторами не является магистральным соединением.

Теперь рассмотрим пример 9.4, демонстрирующий новую конфигурацию, где магистральное соединение разрешено. В данном случае коммутатор SW1 настроен командой `switchport mode dynamic desirable`, требующей от коммутатора начать процесс переговоров, а не ждать их от другого устройства. Как только команда будет введена, появятся регистрационные сообщения, свидетельствующие об отключении, а затем о включении интерфейса, как обычно происходит при переходе интерфейса из режима доступа в режим магистрального канала.

Пример 9.4. Изменение режима коммутатора SW1 с `dynamic auto` на `dynamic desirable`

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface gigabit 0/1
SW1(config-if)# switchport mode dynamic desirable
SW1(config-if)# ^Z
SW1#
01:43:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed
state to down
01:43:49: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed
state to up
SW1# show interfaces gigabit 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
```

```
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
! строки опущены для краткости

! Следующая команда раньше выводила одну пустую строку, а теперь ее вывод
! содержит информацию об одной действующей магистрали.

SW1# show interfaces trunk

Port    Mode         Encapsulation   Status   Native vlan
Gi0/1   desirable 802.1q           trunking 1

Port    Vlans allowed on trunk
Gi0/1   1-4094

Port    Vlans allowed and active in management domain
Gi0/1   1-3

Port    Vlans in spanning tree forwarding state and not pruned
Gi0/1   1-3

SW1# show vlan id 2
VLAN Name                 Status     Ports
---- -------------------- ---------- -------------------------------------
2    Freds-vlan           active     Fa0/13, Fa0/14, G0/1

VLAN Type  SAID   MTU  Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
---- ----- ------ ---- ------ ------ -------- --- -------- ------ -------
2    enet  100010 1500 -      -      -        -   -        0      0

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------
```

Для проверки того, что магистральное соединение теперь работает, в середине примера 9.4 введена команда `show interfaces switchport`. Обратите внимание: в ее выводе по-прежнему отображаются административные параметры, демонстрирующие введенные в конфигурацию значения, но наряду с ними отображаются рабочие настройки, благодаря чему можно узнать, какие действия в настоящее время выполняются коммутатором. В данном случае коммутатор SW1 находится в рабочем режиме магистрального канала с применением инкапсуляции `dot1Q`.

В конце примера приведен вывод команды `show interfaces trunk`, который отображает интерфейс G0/1, подтверждая, что теперь он работает как магистраль. Смысл вывода этой команды обсуждается в следующем разделе.

Для подготовки к экзаменам следует быть готовым интерпретировать вывод команды `show interfaces switchport`, разбираться в том, как по ее выводу определять заданный административный режим, и знать, должно ли в канале быть создано магистральное соединение, если применяются указанные настройки. В табл. 9.2 перечислены сочетания административных режимов магистрального соединения и ожидаемые режимы работы (магистральный или доступа), устанавливаемые в результате

применения заданных параметров. В левой части таблицы указаны административные режимы, используемые коммутатором на одном конце канала, а в верхней части — административные режимы, заданные для коммутатора на другом конце канала.

Ключевая тема

Таблица 9.2. Ожидаемый рабочий режим магистрали на основании параметров административных режимов

Административный режим	access	dynamic auto	trunk	dynamic desirable
access	access	access	Не используется *	access
dynamic auto	access	access	trunk	trunk
trunk	Не используется *	trunk	trunk	trunk
dynamic desirable	access	trunk	trunk	trunk

* Когда коммутатор на одном конце находится в режиме access, а на другом конце в режиме trunk, возникают проблемы. Избегайте такой комбинации.

Компания Cisco рекомендует отключать переговоры магистральных каналов на большинстве портов для повышения их зашиты. Большинство портов на большинстве коммутаторов используется для подключения пользователей. Не забывайте, переговоры DTP можно отключить в целом с помощью подкоманды интерфейса switchport nonegotiate.

Контроль сетей VLAN, поддерживаемых на магистральном канале

Список разрешенных сетей VLAN (allowed VLAN list) — это механизм, позволяющий сетевым инженерам административно отключать сети VLAN от магистрального канала. Стандартно коммутаторы включают в список разрешенных сетей VLAN каждой магистрали все возможные сети VLAN (от VLAN 1 до VLAN 4094). Однако впоследствии инженер может сократить количество сетей VLAN, которым разрешено использовать магистральный канал. Для этого используется следующая подкоманда интерфейса:

```
switchport trunk allowed vlan {add | all | except | remove} список_vlan
```

Эта команда предоставляет удобный способ добавления и удаления сетей VLAN из списка разрешенных. В частности, параметр add позволяет коммутатору добавлять сети VLAN к существующему списку разрешенных сетей VLAN, а с помощью параметра remove можно удалить сети VLAN из существующего списка коммутатора. Под параметром all подразумеваются все сети VLAN, поэтому он может применяться для переустановки коммутатора в стандартную конфигурацию (разрешающую применение магистрали для сетей VLAN от 1 до 4094). С другой стороны, параметр except является довольно сложным; он позволяет добавить к списку все сети VLAN, не указанные в команде. Например, выполнение подкоманды интерфейса switchport trunk allowed vlan except 100-200 добавит к существующему списку разрешенных сетей сети VLAN 1–VLAN 99 и сети VLAN 201–VLAN 4094.

Кроме списка разрешенных сетей VLAN, коммутатор может руководствоваться другими причинами для запрета передачи через определенную магистраль трафика конкретной сети VLAN. Все пять причин запрета прохождения трафика перечислены ниже.

Причины невозможности передачи трафика сети VLAN по магистральному каналу

Ключевая тема

- Сеть VLAN удалена из списка *разрешенных сетей VLAN* для магистрального канала.
- Сеть VLAN отсутствует в конфигурации коммутатора (как свидетельствует вывод команды `show vlan`).
- Сеть VLAN существует, но административно отключена (командой `shutdown`).
- Сеть VLAN автоматически отсечена протоколом VTP.
- Экземпляр STP сети VLAN перевел магистральный интерфейс в состояние блокировки.

ВНИМАНИЕ!

Последние две причины не рассматриваются в этой книге, но упоминаются здесь для порядка.

Первая причина (список разрешенных сетей VLAN) уже упоминалась в этом разделе, а теперь обсудим две следующие. Если у коммутатора нет информации о существовании какой-то сети VLAN (т.е. команды `vlan` *`идентификатор_vlan`* в конфигурации коммутатора нет, что подтверждает вывод команды `show vla`), то он не будет перенаправлять фреймы этой сети VLAN ни по какому интерфейсу. Кроме того, сеть VLAN может существовать в конфигурации коммутатора, но быть административно отключенной либо глобальной командой конфигурации `shutdown vlan` *`идентификатор_vlan`*, либо командой `shutdown` в режиме конфигурации VLAN. Когда сеть VLAN отключена, коммутатор больше не будет перенаправлять ее фреймы даже по магистральным каналам. В результате коммутаторы не перенаправляют фреймы несуществующих или отключенных сетей VLAN ни по одному из магистральных каналов.

В этой книге есть смысл перечислить причины невозможности передачи трафика сети VLAN по магистральному каналу: команда `show interfaces trunk` выводит список идентификаторов VLAN в том же порядке, на основании тех же причин. Вывод указанной команды содержит продолжение в виде трех списков сетей VLAN, поддерживаемых магистральным каналом.

- Сети VLAN, разрешенные на магистральном канале (стандартно 1–4094).
- Сети VLAN из первой группы, настроенные и активные (не отключенные).
- Сети VLAN из второй группы, не отсеченные протоколом VTP и не блокированные протоколом STP.

Чтобы получить представление об этих трех списках в выводе команды `show interfaces trunk`, рассмотрим пример 9.5, демонстрирующий варианты запрета передачи трафика сетей VLAN через магистральный канал по разным причинам. Вывод указанной команды получен на коммутаторе SW1 (см. рис. 9.12) после настройки конфигурации в соответствии с предыдущими примерами этой главы. Другими словами, сети VLAN 1–3 существуют в конфигурации коммутатора SW1 и не отключены. Магистральное соединение между коммутаторами SW1 и SW2 находится в рабочем состоянии. Затем в ходе выполнения данного примера на коммутаторе SW1 настраиваются следующие параметры конфигурации.

Этап 1 Настройка сети VLAN 4

Этап 2 Отключение сети VLAN 2

Этап 3 Удаление сети VLAN 3 из списка разрешенных сетей VLAN магистрального канала

Пример 9.5. Списки разрешенных и активных сетей VLAN

```
! Три списка сетей VLAN в выводе следующей команды показывают разрешенные
! сети VLAN (1-4094), разрешенные и активные сети VLAN (1-3) и
! разрешенные, активные, неотсеченные и перенаправляемые STP
! сети VLAN (1-3).
SW1# show interfaces trunk

Port    Mode         Encapsulation   Status    Native vlan
Gi0/1   desirable 802.1q           trunking 1

Port    Vlans allowed on trunk
Gi0/1   1-4094

Port    Vlans allowed and active in management domain
Gi0/1   1-3

Port    Vlans in spanning tree forwarding state and not pruned
Gi0/1   1-3

! После этого в конфигурации коммутатора выполняется настройка новой сети
! VLAN 4, сеть VLAN 2 отключается, а сеть VLAN 3 удаляется из списка
! разрешенных сетей VLAN для этой магистрали.
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# vlan 4
SW1(config-vlan)# vlan 2
SW1(config-vlan)# shutdown
SW1(config-vlan)# interface gi0/1
SW1(config-if)# switchport trunk allowed vlan remove 3
SW1(config-if)# ^Z

! Три списка сетей VLAN в выводе следующей команды показывают разрешенные
! сети VLAN (1, 2 и 4-4094), разрешенные и активные сети VLAN (1 и 4) и
! разрешенные, активные, неотсеченные и перенаправляемые протоколом STP
! сети VLAN (1 и 4).

SW1# show interfaces trunk
Port    Mode         Encapsulation   Status    Native vlan
Gi0/1   desirable 802.1q           trunking 1

! Далее сеть VLAN 3 исключается, поскольку она была удалена из списка
! разрешенных сетей VLAN.
Port    Vlans allowed on trunk
Gi0/1   1-2,4-4094

! Сеть VLAN 2 исключается, поскольку она отключена. Сети VLAN 5-4094
! исключаются, поскольку на коммутаторе SW1 они не настроены.
Port    Vlans allowed and active in management domain
Gi0/1   1,4

Port    Vlans in spanning tree forwarding state and not pruned
Gi0/1   1,4
```

Обзор

Резюме

- Локальная сеть (LAN) объединяет все устройства в том же широковещательном домене.
- Без виртуальных сетей коммутатор полагает, что все его интерфейсы находятся в том же широковещательном домене.
- Коммутатор VLAN может настроить часть интерфейсов на один широковещательный домен, а часть на другой, создав в результате два широковещательных домена. Эти созданные коммутатором индивидуальные широковещательные домены и являются виртуальными локальными сетями (VLAN).
- Ниже приведен список наиболее распространенных причин создания меньших широковещательных доменов (сетей VLAN).
  - Сокращение дополнительных затрат процессоров всех устройств за счет сокращения количества устройств, получающих каждый широковещательный фрейм.
  - Улучшение защиты за счет сокращения количества хостов, получающих копии фреймов при их лавинной рассылке коммутатором (широковещание, групповая передача и одноадресатные фреймы с неизвестным получателем).
  - Улучшение защиты хостов, пересылающих важные данные, за счет их помещения в отдельную сеть VLAN.
  - Возможность более гибкого объединения пользователей в группы (например, по отделам) вместо физического разделения по местоположению.
  - Упрощение поиска проблемы в сети, поскольку большинство проблем локализуется в области набора устройств, формирующих широковещательный домен.
  - Сокращение дополнительных затрат на работу протокола распределенного связующего дерева (STP) за счет ограничения VLAN одним коммутатором доступа.
- Настройка сети VLAN с одним коммутатором требует немного усилий: достаточно настроить каждый порт так, чтобы указать ему номер VLAN, к которой он принадлежит.
- Когда сети VLAN используются в сетях с несколькими соединенными между собой коммутаторами, на каналах связи между ними применяется магистральное соединение VLAN.
- Магистральное соединение VLAN подразумевает использование коммутаторами процесса назначения тегов VLAN, когда передающий коммутатор добавляет к фрейму другой заголовок перед его передачей по магистральному каналу. Этот дополнительный заголовок включает поле идентификатора VLAN (VLAN ID), позволяющего передающему коммутатору ассоциировать фрейм с конкретной сетью VLAN, а получающему коммутатору узнать, к какой именно VLAN принадлежит данный фрейм.

- В последние годы компания Cisco использует два разных протокола магистральных соединений: протокол межкоммутаторных соединений (ISL) и протокол 802.1Q стандарта IEEE.
- Для каждого магистрального канала стандарт 802.1Q определяет также один специальный идентификатор VLAN, обозначающий собственную сеть VLAN (стандартно это VLAN 1).
- При создании территориальной локальной сети, содержащей много сетей VLAN, требуется обеспечить всем устройствам возможность передавать данные на все остальные устройства.
- Окончательное решение подразумевает передачу функций маршрутизации аппаратным средствам коммутатора LAN. Производители уже довольно давно начали объединять аппаратные и программные средства коммутаторов уровня 2 с маршрутизаторами уровня 3, выпуская коммутаторы уровня 3 (они же многоуровневые коммутаторы). Коммутаторы уровня 3 могут быть настроены так, чтобы действовать только как коммутаторы уровня 2 или коммутаторы уровня 2 с маршрутизаторами уровня 3.
- Последовательность настройки конфигурации VLAN и назначения интерфейсов.

  Этап 1 Чтобы настроить конфигурацию новой сети VLAN, выполните следующие действия:

  A. В режиме настройки конфигурации введите глобальную команду конфигурации `vlan` *`идентификатор_vlan`* для создания сети VLAN и перейдите в режим настройки конфигурации сети VLAN.

  B. (Необязательно.) Чтобы присвоить сети VLAN имя, введите подкоманду VLAN `name` *`имя`*. Если этого не сделать, именем VLAN будет `VLAN`*`ZZZZ`*, где *`ZZZZ`* — десятичный идентификатор из четырех цифр

  Этап 2 Для каждого интерфейса доступа (интерфейса, принадлежащего не магистральному каналу, а отдельной сети VLAN) выполните следующие действия:

  A. Используя команду `interface`, перейдите в режим конфигурации каждого настраиваемого интерфейса.

  B. Используя подкоманду интерфейса `switchport access vlan` *`идентификатор_vlan`*, укажите номер VLAN, связанной с данным интерфейсом.

  C. (Необязательно). Чтобы отключить магистральное соединение на том же интерфейсе и запретить переговоры о создании магистрального канала, используйте подкоманду интерфейса `switchport mode access`

- Протокол создания магистралей VLAN — это собственный протокол компании Cisco, выполняющийся на ее коммутаторах. Он анонсирует каждую сеть VLAN, настроенную на одном коммутаторе командой `vlan` *`номер`*, чтобы о ней узнали все остальные коммутаторы в территориальной локальной сети.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Какой из следующих терминов, применяемых в локальной сети, наиболее соответствует термину *сеть VLAN*?

   А) Домен коллизий.

   Б) Широковещательный домен.

   В) Домен подсети.

   Г) Отдельный коммутатор.

   Д) Магистраль.

2. Предположим, имеется коммутатор с тремя сетями VLAN. Сколько требуется подсетей IP при условии, что на всех хостах во всех сетях VLAN должны применяться протоколы TCP/IP?

   А) 0.

   Б) 1.

   В) 2.

   Г) 3.

   Д) Об этом нельзя судить на основании лишь предоставленной информации.

3. Коммутатор SW1 посылает фрейм коммутатору SW2 по магистрали, использующей протокол 802.1Q. Какой из ответов описывает процесс изменения или добавления коммутатором SW1 заголовка фрейма Ethernet перед его перенаправлением на коммутатор SW2?

   А) Добавляет 4-байтовый заголовок и изменяет MAC-адрес.

   Б) Добавляет 4-байтовый заголовок и не изменяет MAC-адрес

   В) Инкапсулирует первоначальный фрейм в совершенно новый заголовок Ethernet

   Г) Все ответы неверные.

4. Между двумя коммутаторами Ethernet существует магистральный канал 802.1Q. Какой из ответов наиболее точно определяет фреймы, в которые не включается заголовок 802.1Q?

   А) Фреймы в собственной сети VLAN (только один).

   Б) Фреймы сетей VLAN расширенного диапазона.

   В) Фреймы сети VLAN 1 (не настраиваемой)

   Г) Фреймы всех собственных сетей VLAN (когда разрешено несколько).

5. Предположим, получено такое указание, что коммутатор 1 настроен с параметром `dynamic auto` для создания магистрали на интерфейсе Fa0/5, который подключен к коммутатору 2. Необходимо настроить конфигурацию коммутатора 2. Какая из следующих настроек для магистрального соединения может обеспечить работу магистрали? (Выберите два ответа.)

   А) Перевод магистрали в режим `on`.

   Б) Параметр `dynamic auto`.

   В) Параметр `dynamic desirable`.

   Г) Параметр `access`.

   Д) Все ответы неверные.

6. Коммутатор только что получен от корпорации Cisco. Еще не проводилось никаких настроек сетей VLAN, протокола VTP или любой другой конфигурации. Инженер переходит в режим настройки конфигурации и вводит команду `vlan 22`, за которой следует команда `name Hannahs-VLAN`. Какое из следующих утверждений является истинным?

   А) В выводе команды `show vlan brief` отображается сеть VLAN 22.

   Б) В выводе команды `show running-config` отображается сеть VLAN 22.

   В) Сеть VLAN 22 не создается в этом процессе.

   Г) Сеть VLAN 22 не существует в этом коммутаторе до тех пор, пока в нее не добавлено ни одного интерфейса.

7. Какая из следующих команд позволяет получить информацию о состоянии интерфейсов коммутатора, т.е. работают ли они в настоящий момент как магистральные каналы VLAN? (Выберите два ответа).

   А) `show interfaces`

   Б) `show interfaces switchport`

   В) `show interfaces trunk`

   Г) `show trunks`

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 9.3.

Таблица 9.3. Ключевые темы главы 9

Элемент	Описание	Страница
Рис. 9.2	Создание двух широковещательных доменов с использованием одного коммутатора и сети VLAN	285
Список	Причины применения сетей VLAN	286
Рис. 9.5	Магистральное соединение VLAN между двумя коммутаторами	288
Рис. 9.6	Заголовок магистрального соединения по стандарту 802.1Q	289
Рис. 9.9	Маршрутизация между двумя сетями VLAN с использованием магистрального канала на маршрутизаторе	292
Рис. 9.10	Маршрутизация между сетями VLAN с использованием коммутатора уровня 3	294
Список	Последовательность настройки конфигурации VLAN и назначения интерфейсов	294
Табл. 9.1	Параметры команды switchport mode, определяющие административный режим магистрали	300
Табл. 9.2	Ожидаемый рабочий режим магистрали на основании параметров административных режимов	304
Список	Причины невозможности передачи трафика сети VLAN по магистральному каналу	305

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

802.1Q, магистральный канал (trunk), административный режим магистрали (trunking administrative mode), рабочий режим магистрали (trunking operational mode), VLAN, VTP, прозрачный режим VTP (VTP transparent mode), коммутатор третьего уровня (Layer 3 switch), интерфейс доступа (access interface), магистральный интерфейс (trunk interface)

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 9.4 приведен список команд конфигурации, а в табл. 9.5 пользовательские команды главы. Фактически команды стоит запомнить, чтобы лучше понять содержимое главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспомнить команду.

Таблица 9.4. Команды конфигурации главы 9

Команда	Описание
`vlan` *`идентификатор_vlan`*	Глобальная команда конфигурации, позволяющая создать сеть VLAN и перевести интерфейс командной строки в режим настройки конфигурации сети VLAN
`Name` *`имя_vlan`*	Подкоманда сети VLAN, позволяющая присвоить имя сети VLAN
`[no] shutdown`	Подкоманда режима VLAN, позволяющая включить (`no shutdown`) или отключить (`shutdown`) сеть VLAN
`[no] shutdown vlan` *`идентификатор_vlan`*	Глобальная команда конфигурации, аналогичная подкоманде режима VLAN `[no] shutdown`
`vtp mode {server | client | transparent | off}`	Глобальная команда конфигурации, определяющая режим VTP
`switchport mode {access | dynamic {auto | desirable} | trunk}`	Подкоманда интерфейса, задающая административный режим магистрального соединения на интерфейсе
`switchport trunk allowed vlan {add | all | except | remove}` *`список_vlan`*	Подкоманда интерфейса, определяющая список разрешенных сетей VLAN

Окончание табл. 9.4

Команда	Описание
`switchport access vlan` *идентификатор_vlan*	Подкоманда интерфейса, применяемая для статической настройки интерфейса при подключении к одной указанной сети VLAN
`switchport trunk encapsulation {dot1q | isl | negotiate}`	Подкоманда интерфейса, определяющая тип используемого магистрального соединения с учетом того, задано ли магистральное соединение в конфигурации или согласовано
`switchport trunk native vlan` *идентификатор_vlan*	Подкоманда интерфейса, определяющая собственную сеть VLAN для порта магистрального канала
`switchport nonegotiate`	Подкоманда интерфейса, запрещающая согласование при создании магистрали VLAN

Таблица 9.5 Пользовательские команды главы 9

Команда	Описание
`show interfaces` *идентификатор_интерфейса* `switchport`	Выводит информацию о любом интерфейсе, относящуюся к административным настройкам и рабочему состоянию
`show interfaces` *идентификатор_интерфейса* `trunk`	Выводит информацию обо всех действующих магистралях (но не о других интерфейсах), включая список сетей VLAN, трафик которых может быть перенаправлен по данной магистрали
`show vlan [brief | id` *идентификатор_vlan* `| name` *имя_vlan* `| summary]`	Выводит информацию о сети VLAN
`show vlan [`*vlan*`]`	Отображает информацию о сети VLAN
`show vtp status`	Выводит информацию о конфигурации протокола VTP и о состоянии

Ответы на контрольные вопросы:

1 Б. **2** Г. **3** Б. **4** А. **5** А и В. **6** А и Б. **7** Б и В.

Глава 10

Поиск и устранение неисправностей на коммутаторах Ethernet

Эта глава посвящена процессам проверки, а также поиску и устранению неисправностей. Под *проверкой* (verification) подразумевается процесс подтверждения того, что сеть работает так, как задумано. Под *поиском и устранением неисправностей* (troubleshooting) подразумевается последующий процесс (когда уже установлено, что сеть работает не так, как задумано) выявления реальных причин проблем и их устранения.

Со временем на экзаменах CCENT и CCNA встречается все больше вопросов на проверку и устранение неисправностей. Каждый из этих вопросов требует применения общего знания сети к конкретным проблемам, а не просто теоретического ответа с перечислением списка запомненных фактов.

Чтобы помочь подготовиться к ответам на вопросы о поиске и устранении неисправностей, в этой книге, а также в книге по ICND2 данной теме посвящается как отдельные разделы, так и целые главы. В них не только приведена конфигурация и примеры вывода разных команд `show`, но и обсуждается применение разных команд для проверки происходящего, и если это не то, что ожидалось, то какими способами искать первопричину проблемы.

В этой главе обсуждается множество тем, большинство из которых уже было затронуто в главах 6–9. Глава начинается с обсуждения концепции поиска и устранения неисправностей в сети, поскольку это первая глава книги, полностью посвященная данной теме. Далее рассматриваются следующие четыре ключевые темы, весьма важные при проверке, поиске и устранении неисправностей локальных сетей Ethernet.

- Анализ топологии локальной сети с использованием протокола CDP.
- Анализ состояния интерфейса коммутатора.
- Прогноз перенаправления фреймов коммутаторами.
- Анализ сетей VLAN и магистральных каналов VLAN.

В этой главе рассматриваются следующие экзаменационные темы

Технологии коммутации сетей LAN

Настройка и проверка магистрального соединения на коммутаторах Cisco.

Протокол DTP.

Автопереговоры.

Защита сетевых устройств

Настройка и проверка средств защиты порта коммутатора.

Статические и динамические.

Реакция при нарушении защиты.

Отключение из-за ошибки.

Отключение.

Ограничение.

Поиск и устранение неисправностей

Поиск неисправностей и решение проблем сетей VLAN.

Идентификация настроенных сетей VLAN.

Исправление принадлежности порта.

Настройка IP-адреса.

Поиск неисправностей и решение проблем магистрального соединения на коммутаторах Cisco.

Исправление состояния магистрального канала.

Исправление конфигурации инкапсуляции.

Исправление разрешенных VLAN.

Поиск неисправностей и решение проблем уровня 1.

Фреймирование.

CRC.

Карлики.

Гиганты.

Отброшенные пакеты.

Запоздалые коллизии.

Ошибки ввода и вывода.

Основные темы

Принципы проверки сетей и устранения неисправностей

ВНИМАНИЕ!

Информация, представленная в этом разделе, поможет приобрести полезные навыки поиска и устранения неисправностей в сетях. Специфические советы и методики, представленные здесь, не связаны с какой-либо конкретной темой экзаменов CCENT и CCNA.

Для успешной сдачи экзамена CCENT и CCNA недостаточно только теоретических знаний, нужны также некоторые практические навыки, чтобы ответить на сложные прикладные вопросы. Следует также отметить, что уровень сложности вопросов в сертификационном экзамене разный. В текущем разделе сначала рассмотрены различные типы вопросов, которые могут быть на экзамене CCNA, а потом даны общие комментарии по методам поиска и устранения неисправностей.

Во введении к этой книге кратко описано несколько разных типов экзаменационных вопросов, рассмотренных в данной главе: *лабораторные работы на эмуляторах оборудования* (Simulated lab — Sim), *симлеты* (simlet) и *многовариантный выбор* (Multiple Choice — MC).

Лабораторные работы и симлеты используют эмулятор, моделирующий интерфейс командной строки маршрутизаторов и коммутаторов. В лабораторных работах требуется найти проблему в конфигурации и устранить ее. В симлетах требуется проверить текущую работу сети, а затем ответить на вопросы с многовариантным выбором о ее работе. Вопросы с многовариантным выбором — это просто вопросы с несколькими ответами, из которых следует выбрать правильный.

ВНИМАНИЕ!

С типами экзаменационных вопросов CCENT и CCNA можно ознакомиться на веб-сайте `www.cisco.com/web/learning/wwtraining/certprog/training/cert_exam_tutorial.html`.

Процесс упорядоченного поиска и устранения неисправностей

В день экзамена у вас будет одна цель: правильно ответить на достаточно много вопросов, чтобы сдать экзамен. Для этого перед экзаменом следует правильно организовать сам процесс ответов на вопросы. При подготовке к экзамену следует многому научиться и обдумать процесс поиска неисправностей, чтобы в день экзамена быть готовым решать проблемы быстро.

Процесс упорядоченного поиска и устранения неисправностей хорош при подготовке к экзамену, когда времени много, — он поможет лучше разобраться в сетевых технологиях и подготовиться к экзамену. В этой книге рассмотрены многие методы поиска и устранения неисправностей, тем не менее такие методы не являются самоцелью, поэтому не нужно запоминать их на память. Методы поиска и устранения неисправностей представляют собой в основном инструмент обучения, а также имеют дополнительный плюс — помогают подготовиться к самым сложным вопросам сертификационного экзамена.

В этом разделе описан некоторый обобщенный процесс поиска и устранения неисправностей в сетях. По мере изучения материала данной книги этот процесс будет периодически повторяться, но уже для определенных сетевых технологий, например для IP-маршрутизации. Три основных этапа процесса поиска и устранения неисправностей подробно описаны ниже.

Этап 1 Прогноз и анализ нормального поведения

Нужно представить себе, что должно происходить, когда сеть работает правильно, на основании документации, конфигурационных команд и результатов выполнения команд `show` и `debug`

Этап 2 Изоляция проблемы

Необходимо определить, как далеко по пути следования пакета или фрейма распространяется проблема и где начинается нормальная работа сети, опять же на основании документации, конфигурационных команд и результатов выполнения команд `show` и `debug`

Этап 3 Анализ источника проблемы

Необходимо идентифицировать глубинные причины проблемы, которая была выявлена на предыдущем этапе. В частности, следует подумать над тем, какие именно действия помогут решить проблему

Следование описанному выше процессу требует достаточно обширных знаний и навыков. Чтобы успешно искать отказы в сети, следует хорошо помнить теоретические основы работы компьютерных сетей, уметь интерпретировать результаты различных команд `show` в разных ситуациях и режимах работы сети. В процессе поиска неисправностей понадобятся дополнительные инструменты, такие как команды `ping` и `traceroute`, которые помогут локализовать и изолировать проблемный участок. Кроме того, нужно уметь мыслить широко и обладать разнообразными знаниями, чтобы учесть все факторы, которые могут влиять на какой-либо компонент сети.

Рассмотрим, например, следующую проблему в сети, показанной на рис. 10.1. Компьютеры PC1 и PC2 предположительно находятся в той же сети VLAN (10). Команда `ping 10.1.1.2` на компьютере PC1 иногда срабатывает, а иногда нет.

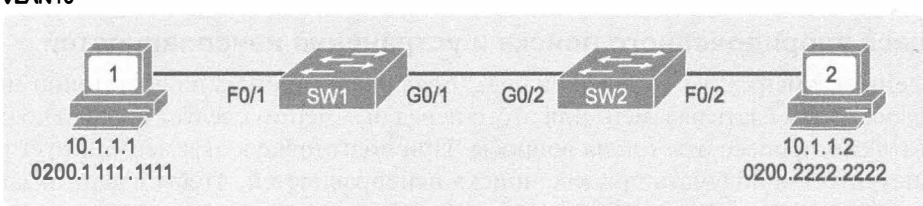


Рис. 10.1. Пример сети с проблемной командой `ping`

Так как же приступить к устранению этой проблемы? Если есть сомнения в правильности рисунка, посмотрите на вывод команды `show` и проверьте топологию сети. Подтвердив топологию на основании знания правил коммутации в сети LAN, можно прогнозировать ее нормальное рабочее поведение, т.е. куда должен направиться фрейм, посланный компьютером PC1 компьютеру PC2. Для локализации проблемы можно просмотреть таблицу MAC-адресов на коммутаторе, чтобы прове-

рить интерфейсы, на которые фрейм должен быть перенаправлен. Возможно, окажется, что подключенный к компьютеру PC2 интерфейс неисправен.

Даже выяснив, что интерфейс действительно неисправен, первопричина этого остается неизвестной. Как можно устранить проблему, не зная ее причины? В данном конкретном случае следует рассмотреть все возможные причины неисправности интерфейса: от отключения кабеля и электрических помех до отключения интерфейса средствами защиты порта. Команда `show` может либо подтвердить первопричину проблемы, либо дать некоторую подсказку о ней.

Первая обнаружившаяся проблема примера — используется простая сеть LAN с одной подсетью и без маршрутизации IP. Но в большинстве экзаменационных вопросов будет несколько подсетей IP с маршрутизаторами, перенаправляющими пакеты IP между хостами. В этих случаях процесс поиска неисправности зачастую начинается с анализа процесса маршрутизации уровня 3 при перенаправлении пакетов IP.

Например, пользователь компьютера PC1 (рис. 10.2) может ввести в веб-браузере адрес `www.example.com` и подключиться к веб-серверу (справа на рисунке). Однако попытка просмотра веб-страницы терпит неудачу. Пользователь обращается в службу технической поддержки, и сетевому инженеру предстоит решить проблему.

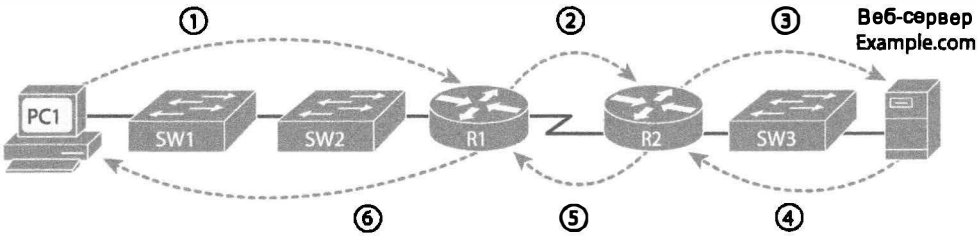


Рис. 10.2. Изоляция проблемы третьего уровня

Сетевой инженер может начать анализ с первых задач, необходимых для успешного просмотра веб-страниц. Например, он может проверить, способен ли компьютер PC1 преобразовать имя хоста (`www.example.com`) в правильный IP-адрес, используемый сервером, который показан справа. Далее процесс изоляции проблемы на третьем уровне для технологии IP может следовать шести указанным ниже этапам. Описание этапов, показанных на рис. 10.2, приведено ниже.

Этап 1 Компьютер PC1 пересылает пакет своему стандартному шлюзу (маршрутизатору R1), поскольку IP-адрес получателя (веб-сервера) находится в другой подсети

Этап 2 Маршрутизатор R1 пересылает пакет маршрутизатору R2 согласно записям в своей таблице маршрутизации

Этап 3 Маршрутизатор R2 пересылает пакет веб-серверу согласно записям в своей таблице маршрутизации

Этап 4 Веб-сервер отправляет ответный пакет компьютеру PC1 через свой стандартный шлюз (маршрутизатор R2)

Этап 5 Маршрутизатор R2 пересылает ответный пакет для компьютера PC1 маршрутизатору R1 согласно записям в своей таблице маршрутизации

Этап 6 Маршрутизатор R1 пересылает пакет компьютеру PC1 согласно записям в своей таблице маршрутизации

Большинство инженеров разделяют решение проблемы на этапы, как в этом списке, анализируя шаг за шагом пути на третьем уровне в обоих направлениях. Этот процесс позволяет получить представление о первой попытке изоляции проблемы. Когда анализ покажет, на каком этапе возникла проблема, можно перейти к изучению подробностей. Поскольку в данном случае процесс изоляции проблемы на уровне 3 обнаруживает проблемы на этапах 1, 3, 4 и 6, первопричина, вероятно, связана с Ethernet.

Представим себе ситуацию, когда анализ третьего уровня выявил, что компьютер PC1 не может отправить пакет своему *стандартному шлюзу* (default gateway), т.е. проблемы появились на первом этапе (см. рис. 10.2). Чтобы еще более четко изолировать проблему и ее причины, инженеру потребуется собрать следующую информацию:

- определить MAC-адрес компьютера PC1 и интерфейса локальной сети маршрутизатора R1;
- идентифицировать используемые интерфейсы коммутаторов SW1 и SW2;
- определить состояние всех интерфейсов;
- определить используемые сети VLAN;
- проверить пересылку фрейма от компьютера PC1 к маршрутизатору R1, непосредственно для MAC-адреса последнего.

Собрав и проанализировав все вышеперечисленные факты, инженер, скорее всего, найдет, в чем причина проблемы, и исправит ее.

Советы по поиску и устранению неисправностей

В текущей версии экзаменов ICND1 и ICND2 тема поиска и устранения неисправностей распространяется на оба экзамена. Тем не менее в текущей версии экзаменов (100-101 ICND1 и 200-101 ICND2) экзамен ICND2 уделяет этой теме куда больше внимания, чем экзамен ICND1. В результате данная книга посвящает поиску и устранению неисправностей только одну главу (данную) и несколько разделов в остальных главах. Второй том данного академического издания уделяет поиску и устранению неисправностей куда больше внимания.

В остальной части этой главы рассматриваются четыре главных темы поиска и устранения неисправностей в локальных сетях Ethernet. Из них только первая тема, протокол обнаружения устройств (CDP), представляет совершенно новый материал. Другие три темы уже знакомы: речь идет о поиске и устранении неисправностей. Вот эти темы.

- *Протокол обнаружения устройств Cisco* (Cisco Discovery Protocol — CDP). Используется для проверки существующей документации сети, изучения сетевой топологии и оценки нормальной работы сети.
- *Анализ состояния интерфейса.* Все интерфейсы должны быть в рабочем состоянии, чтобы коммутатор мог пропустить через себя трафик. Сетевой инженер должен быть в состоянии определить, работает ли интерфейс, и если нет, то найти причины отказа.

- *Анализ маршрута коммутации фреймов* предполагает, что специалист может проанализировать таблицу MAC-адресов и оценить, по какому маршруту (или через какой интерфейс) коммутатор перешлет фрейм.
- *Анализ сети VLAN и магистрального соединения*. Основное внимание в последнем разделе уделяется коммутаторам уровня 2 и проблемам с сетями VLAN, а также магистральными каналами VLAN.

Анализ топологии локальной сети с помощью протокола обнаружения устройств Cisco

Собственный *протокол обнаружения устройств Cisco* (Cisco Discovery Protocol — CDP) позволяет получить базовую информацию о соседних маршрутизаторах и коммутаторах, даже не зная пароль для доступа к ним. Чтобы получить информацию, маршрутизаторы и коммутаторы рассылают сообщения CDP через все свои работающие интерфейсы. Такие сообщения содержат информацию об устройстве, которое его отправило; о других устройствах маршрутизатор или коммутатор узнает из принимаемых им аналогичных сообщений протокола CDP.

Как обычно, компания Cisco создала протокол CDP для удовлетворения потребностей своих клиентов. Впоследствии IEEE стандартизировал *протокол обнаружения устройств уровня канала связи* (Link Layer Discovery Protocol — LLDP), служащий той же цели. Но большинство предприятий, использующих маршрутизаторы и коммутаторы Cisco, применяют протокол CDP, а протокол LLDP держат в резерве, поэтому данная глава сосредоточена исключительно на протоколе CDP, а не LLDP.

С точки зрения процесса поиска и устранения неисправностей в сетях протокол CDP может использоваться для подтверждения или уточнения сетевой документации и схем, а также для обнаружения новых устройств и интерфейсов в сети. Выяснение реальной схемы подключения устройств в сети и усовершенствование схем фактически являются необходимыми этапами перед началом прогнозирования маршрутов трафика в сети.

В среде, поддерживающей многоадресатную передачу данных на канальном уровне (как Ethernet), протокол CDP рассылает многоадресатные фреймы; в других сетевых средах протокол CDP отправляет копию сообщения на любой известный устройству адрес канального уровня. Таким образом, любое устройство, поддерживающее протокол CDP и подключенное к общей среде с другим таким же устройством, может получить информацию о последнем.

Ключевая тема

Информация, получаемая с помощью протокола CDP

- *идентификатор устройства* (device identifier), обычно это название устройства;
- *список адресов* (address list), представляющий собой адрес сетевого и канального уровней;
- *идентификатор порта* (port identifier) — интерфейс дистанционного маршрутизатора или коммутатора на другом конце канала связи, пославший анонс CDP;
- *список возможностей* (capabilities list) описывает тип устройства (например, соседнее устройство — это коммутатор или маршрутизатор);
- *платформа* (platform) показывает модель соседнего устройства.

Исследование информации, полученной с помощью протокола CDP

Протокол CDP поддерживает команды `show`, отображающие информацию о соседних устройствах, о работе протокола CDP и команды конфигурации, отключающие и включающие протокол CDP.

В табл. 10.1 перечислены три команды `show`, отображающие наиболее важную информацию CDP.

Ключевая тема

Таблица 10.1. Варианты команды `show cdp`, используемые для получения информации о смежных устройствах

Команда	Описание
`show cdp neighbors [тип номер]`	Выводит одну строку информации для всех соседних устройств или для устройства, обнаруженного через указанный интерфейс
`show cdp neighbors detail`	Выдает подробную информацию (около 15 строк) для каждого устройства
`show cdp entry название`	Выдает ту же информацию, что и команда `show cdp neighbors detail`, но только для устройства с указанным именем (регистр символов имени имеет значение)

ВНИМАНИЕ!

Маршрутизаторы и коммутаторы Cisco поддерживают те же команды CDP с теми же параметрами и теми же типами вывода.

Следующий пример демонстрирует мощь команды CDP. Вывод нескольких команд `show cdp` в примере 10.1 приведен для топологии сети, представленной на рис. 10.3.

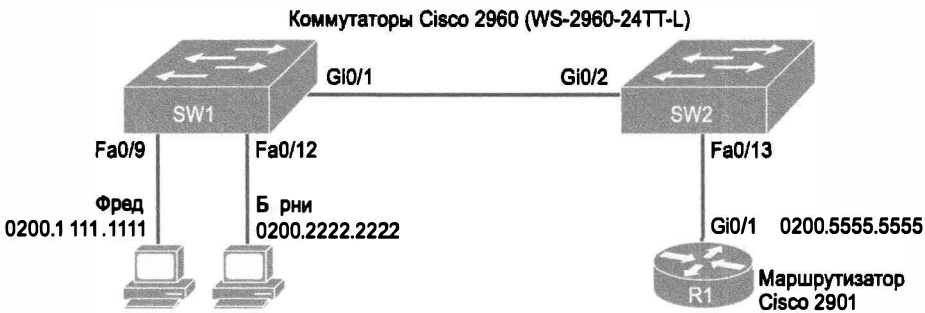


Рис. 10.3. Небольшая сеть, используемая в примерах для протокола CDP

Пример 10.1. Примеры выполнения команд `show cdp` для коммутатора SW2

```
! Команда show cdp neighbors отображает локальные интерфейсы коммутатора
! SW2, а также интерфейсы маршрутизатора R1 и коммутатора SW1 (в столбце
! "port"), наряду с другими подробностями.

SW2# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
```

```
                  P - Phone, D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID      Local Intrfce   Holdtme   Capability  Platform  Port ID
SW1            Gig 0/2         170       S I         WS-C2960- Gig 0/1
R1             Fas 0/13        136       R S I       CISCO2901 Gig 0/1

SW2# show cdp neighbors detail
-------------------------
Device ID: SW1
Entry address(es):
IP address: 172.16.1.1
Platform: cisco WS-C2960-24TT-L, Capabilities: Switch IGMP
Interface: GigabitEthernet0/2, Port ID (outgoing port):
GigabitEthernet0/1
Holdtime : 161 sec
Version :
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version
15.0(1)SE3, RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 30-May-12 14:26 by prod_rel_team

advertisement version: 2
Protocol Hello: OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=00000000FFFFFFFF010221FF00000000000018339D7B0E80FF0000
VTP Management Domain: ''
Native VLAN: 1
Duplex: full
Management address(es):
  IP address: 172.16.1.1

! Авторский комментарий: следующая строка относится к маршрутизатору R1.
-------------------------
Device ID: R1
Entry address(es):
  IP address: 10.1.1.9
Platform: Cisco CISCO2901/K9, Capabilities: Router Switch IGMP
Interface: FastEthernet0/13, Port ID (outgoing port): GigabitEthernet0/1
Holdtime : 127 sec

Version :
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version
15.2(4)M1, RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 26-Jul-12 20:54 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''
Duplex: full
Management address(es):
```

Пример начинается с команды `show cdp neighbors`, выводящей по одной строке о каждом соседе. Каждая строка отображает важнейшую информацию о топологии: имя хоста соседа (столбец `Device ID` (идентификатор устройства)), ин-

терфейсы локального и соседнего устройств (в столбце `Port`). Например, команда `show cdp neighbors` на коммутаторе SW2 выводит запись для коммутатора SW1, где локальным интерфейсом коммутатора SW2 является Gi0/2, а интерфейсом коммутатора SW1 — Gi0/1 (см. рис. 10.3 для справки). Эта команда отображает также платформу, идентифицируя конкретную модель соседнего маршрутизатора или коммутатора. Таким образом, используя даже эту простую информацию, можно либо построить схему сети, как на рис. 10.3, либо подтвердить правильность существующей схемы.

Вывод команды `show cdp neighbors detail` отображает дополнительные подробности, такие как полное название модели коммутатора (WS-2960-24TT-L) и IP-адреса, заданные на соседнем устройстве.

ВНИМАНИЕ!

Команда `show cdp entry` *имя* выводит те же подробности, что и команда `show cdp neighbors detail`, но только для одного соседа, указанного в команде.

Как можно заметить, будучи подключенным на одном устройстве, можно выяснить много информации о соседнем устройстве, что фактически нарушает его защиту. Компания Cisco рекомендует отключать протокол CDP на любом интерфейсе, у которого нет в нем необходимости. Любой порт коммутатора, подключенный к коммутатору, маршрутизатору или телефону IP, должен использовать протокол CDP.

Протокол CDP может быть отключен глобально и на интерфейсе. Подкоманды интерфейса `no cdp enable` и `cdp enable` отключают и включают протокол CDP на интерфейсе, а глобальные команды `no cdp run` и `cdp run` отключают и включают протокол CDP на всем коммутаторе.

Исследование состояния протокола CDP

Протокол CDP определяет сообщения, передаваемые между устройствами. Коммутаторы Cisco поддерживают несколько команд, выводящих информацию о состоянии и другие статистические данные о работе протокола CDP, как показано в табл. 10.2.

Таблица 10.2. Команды для проверки работы протокола CDP

Команда	Описание
`show cdp`	Показывает, включен ли протокол CDP глобально на устройстве, а также какие таймеры *обновлений* (update) и *хранения* информации (holdtime) используются
`show cdp interface` [*тип номер*]	Показывает, включен ли протокол CDP на соответствующих интерфейсах, а также какие таймеры *обновлений* (update) и *хранения* информации (holdtime) используются в этих интерфейсах
`show cdp traffic`	Показывает глобальную статистику обновлений CDP, которые были отправлены и получены устройством

В примере 10.2 приведен вывод каждой из команд в табл. 10.2 на основании топологии коммутатора SW2 (см. рис. 10.3).

Пример 10.2. Команда `show cdp`, отображающая состояние протокола CDP

```
SW2# show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled

SW2# show cdp interface FastEthernet0/13
FastEthernet0/13 is up, line protocol is up
    Encapsulation ARPA
    Sending CDP packets every 60 seconds
    Holdtime is 180 seconds

SW2# show cdp traffic
CDP counters :
    Total packets output: 304, Input: 305
    Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
    No memory: 0, Invalid packet: 0,
    CDP version 1 advertisements output: 0, Input: 0
    CDP version 2 advertisements output: 304, Input: 305
```

Анализ состояния интерфейса коммутатора

В этом разделе рассматриваются интерфейсы коммутатора. Процесс их анализа начинается с выяснения того, работает ли каждый интерфейс коммутатора. Неудивительно, что коммутаторы Cisco не используют интерфейсы вообще, если только они находятся в функциональном или рабочем состоянии. Кроме того, интерфейс коммутатора может находиться в рабочем состоянии, но из-за проблем работать неустойчиво.

Данный раздел начинается с рассмотрения кодов состояния интерфейса коммутатора Cisco и их значения, чтобы можно было узнать, работает ли интерфейс. Затем будут описаны несколько необычные случаи, когда интерфейс работает, но неустойчиво.

Коды состояний интерфейсов и причины их неработоспособности

В коммутаторах компании Cisco используются два набора кодов состояний интерфейсов: один набор состоит из двух кодов (кодовых слов) для каждого состояния, и в нем используется то же соглашение о синтаксисе, что и в маршрутизаторах; второй набор включает в себя по одному коду (кодовому слову) для каждого состояния интерфейса. Оба набора кодовых слов помогут определить, работает интерфейс или нет.

Команды `show interfaces` и `show interfaces description` коммутатора выдают двухкомпонентные коды состояний интерфейса точно так же, как и в маршрутизаторах. Такие коды описывают *состояние линии* (line status) и *состояние протокола* (protocol status) интерфейса и указывают соответственно, работает ли уровень 1 (линия) модели OSI и уровень 2 (протокол канального уровня). Для коммутаторов локальных сетей характерно, что команда показывает два одинаковых значения в обоих полях: два раза "`up`" (работает) или два раза "`down`" (не работает).

ВНИМАНИЕ!

В этой книге оба кода обычно представлены в более удобной сокращенной форме. Например, если уровень линии и уровень протокола работают, то состояние интерфейса записывается так: "`up/up`".

Команда `show interfaces status` выводит для каждого интерфейса одну короткую строку с одним кодовым словом его состояния. Такой код состояния интерфейса имеет однозначную привязку к двухкомпонентному традиционному коду состояния порта. Например, команда `show interfaces status` выдает кодовое слово состояния интерфейса "`connected`" (подключен), чтобы указать, что интерфейс находится полностью в рабочем состоянии. Это кодовое слово соответствует двухкомпонентному коду "`up/up`" (работает/работает) в командах `show interfaces` и `show interfaces description`.

Если перечисленные выше команды выдают код состояния интерфейса, отличный от `connect` или `up/up`, то это означает, что коммутатор не будет передавать и принимать фреймы через него. Для каждого из нерабочих состояний интерфейса есть небольшой набор типичных причин неработоспособности. Следует также помнить, что на сертификационном экзамене может быть приведен одно- или двухкомпонентный код состояния из двух наборов сообщений, следовательно, нужно быть готовым к таким вопросам и помнить все коды на память. Различные комбинации кодов состояния интерфейсов и наиболее распространенные причины неработоспособности приведены в табл. 10.3.

Ключевая тема **Таблица 10.3. Коды состояния интерфейсов коммутаторов локальных сетей**

Состояние линии	Состояние протокола	Состояние интерфейса	Типичные причины
Administratively Down	Down	Disabled	В конфигурации интерфейса введена команда `shutdown`
Down	Down	Notconnect	Кабель не подключен; кабель нерабочий; неправильное расположение выводов кабеля; настройки скорости на двух концах соединения не совпадают; устройство на другом конце кабеля отключено физически, введена команда `shutdown`, отключено из за ошибки
Up	Down	Notconnect	Это состояние (up/down) в коммутаторах локальных сетей практически не встречается
Down	Down (err-disabled)	Err-disabled	Зашита порта отключила порт в связи с нарушением режима безопасности
Up	Up	Connected	Интерфейс работает нормально

Большинство причин для состояния `notconnect` уже было рассмотрено ранее. Это, например, неправильная схема расположения выводов кабеля, как упоминалось в главе 2. Одной из особенно трудно обнаруживаемых причин является рассогласование скорости и дуплекса, обсуждаемое в следующем разделе.

Как можно заметить в таблице, неподходящий кабель — это только одна из многих причин состояния `down/down` (или `notconnect` в выводе команды `show interfaces status`). Вполне очевидно, что экзамены Cisco CCENT и CCNA не особенно сосредоточиваются на кабельной проводке. Ниже приведены некоторые из примеров первопричин проблем, связанных с кабелями.

- Установка любого электрического оборудования, даже никак не связанного с сетевым оборудованием, может создать помехи в кабельной проводке и привести к сбоям в канале связи.
- Кабель может быть поврежден, если, например, находится под ковром. Если стул пользователя передавит кабель, то электрический сигнал может ухудшиться.
- Хотя волоконно-оптические кабели не страдают от электромагнитных помех, кто-то может попробовать перемонтировать или переместить оптоволоконный кабель, перегнув его слишком сильно. Слишком сильный изгиб может ухудшить и даже нарушить передачу битов по кабелю.

Из других состояний интерфейса, перечисленных в табл. 10.3, только состояние `up/up` (`connected`) нуждается в более подробном обсуждении. Интерфейс может быть в рабочем состоянии и действительно работать, но не особенно хорошо. Несколько следующих разделов посвящено выяснению того, работает ли интерфейс в состоянии `up/up` правильно или имеет проблемы.

Проблемы при несовпадении дуплексности и скорости интерфейсов

Большинство интерфейсов Ethernet для кабелей UTP поддерживают несколько скоростей, дуплексный и полудуплексный режимы передачи, а также автопереговоры стандарта IEEE (см. главу 6). Эти же интерфейсы могут быть настроены на использование определенной скорости подкомандой интерфейса `speed {10 | 100 | 1000}` и определенного дуплекса подкомандой интерфейса `duplex {half | full}`. После этого коммутатор или маршрутизатор отключает стандартный процесс автопереговоров IEEE на данном интерфейсе.

В выводимой командами `show interfaces` и `show interfaces status` информации отображаются настройки скорости и дуплексность порта (пример 10.3).

Пример 10.3. Отображение настроек скорости и дуплексности интерфейсов коммутаторов

Ключевая тема

```
SW1# show interfaces status
Port   Name               Status       Vlan  Duplex  Speed  Type
Fa0/1                     notconnect   1     auto    auto   10/100BaseTX
Fa0/2                     notconnect   1     auto    auto   10/100BaseTX
Fa0/3                     notconnect   1     auto    auto   10/100BaseTX
Fa0/4                     connected    1     a-full  a-100  10/100BaseTX
Fa0/5                     connected    1     a-full  a-100  10/100BaseTX
Fa0/6                     notconnect   1     auto    auto   10/100BaseTX
Fa0/7                     notconnect   1     auto    auto   10/100BaseTX
Fa0/8                     notconnect   1     auto    auto   10/100BaseTX
Fa0/9                     notconnect   1     auto    auto   10/100BaseTX
Fa0/10                    notconnect   1     auto    auto   10/100BaseTX
Fa0/11                    connected    1     a-full  10     10/100BaseTX
Fa0/12                    connected    1     half    100    10/100BaseTX
Fa0/13                    connected    1     a-full  a-100  10/100BaseTX
Fa0/14                    disabled     1     auto    auto   10/100BaseTX
Fa0/15                    notconnect   3     auto    auto   10/100BaseTX
Fa0/16                    notconnect   3     auto    auto   10/100BaseTX
Fa0/17                    connected    1     a-full  a-100  10/100BaseTX
```

```
Fa0/18                notconnect  1      auto    auto   10/100BaseTX
Fa0/19                notconnect  1      auto    auto   10/100BaseTX
Fa0/20                notconnect  1      auto    auto   10/100BaseTX
Fa0/21                notconnect  1      auto    auto   10/100BaseTX
Fa0/22                notconnect  1      auto    auto   10/100BaseTX
Fa0/23                notconnect  1      auto    auto   10/100BaseTX
Fa0/24                notconnect  1      auto    auto   10/100BaseTX
Gi0/1                 connected   trunk  full    1000   10/100/1000BaseTX
Gi0/2                 notconnect  1      auto    auto   10/100/1000BaseTX

SW1# show interfaces fa0/13
FastEthernet0/13 is up, line protocol is up (connected)
Hardware is Fast Ethernet, address is 0019.e86a.6f8d (bia 0019.e86a.6f8d)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
   reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mbps, media type is 10/100BaseTX
input flow-control is off, output flow-control is unsupported
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:05, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  85022 packets input, 10008976 bytes, 0 no buffer
  Received 284 broadcasts (0 multicast)
  0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 281 multicast, 0 pause input
  0 input packets with dribble condition detected
  95226 packets output, 10849674 bytes, 0 underruns
  0 output errors, 0 collisions, 1 interface resets
  0 unknown protocol drops
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier, 0 PAUSE output
  0 output buffer failures, 0 output buffers swapped out
```

Обе показанные в примере команды могут быть полезны при проверке работоспособности сети, только с помощью команды `show interfaces status` можно обнаружить, как коммутатор определяет настройки скорости и дуплексности канала. В выводимой этой командой информации автосогласование указывается с помощью приставки `a-` для интерфейса, например, надпись `a-full` сигнализирует о том, что дуплексный режим работы был автоматически согласован, а просто `full` — о том, что такой режим был задан вручную в конфигурации интерфейса. В примере 10.3 выделены цветом строки для двух интерфейсов: для порта Fa0/12 настройки канала были указаны вручную, а порт Fa0/13 автоматически согласовал режим работы. Обратите внимание на то, что с помощью команды `show interfaces fa0/13` (без параметра `status`) можно посмотреть, какие параметры дуплексности и скорости используются для интерфейса FastEthernet0/13, но ничего нельзя сказать о том, как они были получены интерфейсом.

Когда стандарт IEEE автоматического согласования характеристик интерфейса поддерживается обоими устройствами на концах канала, они будут устанавливать максимальную поддерживаемую ими скорость. Аналогично оба устройства сначала попробуют установить дуплексный режим работы канала, а если это не удастся, то согласуют полудуплексный. Если же одно из устройств использует автопереговоры, а второе нет, то первое устройство попытается установить режим дуплексности на основании текущей скорости интерфейса. Стандартные комбинации параметров приведены ниже.

Стандартные правила автопереговоров IEEE

Ключевая тема

- Если скорость не известна, будет использоваться скорость 10 Мбит/с и полудуплексный режим.
- Если коммутатор успешно установил скорость без автопереговоров IEEE, только исходя из сигнала в кабеле:
  - При скорости 10 или 100 Мбит/с стандартно используется полудуплекс.
  - При скорости 11 000 Мбит/с стандартно используется полный дуплекс.

ВНИМАНИЕ!

Интерфейсы Ethernet со скоростью 1 Гбит/с и выше всегда используют полный дуплекс.

Хотя автопереговоры хороши, их стандартное поведение способно привести к весьма трудно обнаруживаемой проблеме — *рассогласованию дуплекса* (duplex mismatch). В главе 6 упоминалось о том, что оба устройства могут использовать ту же скорость, поэтому они будут полагать, что канал связи находится в состоянии `up`, но одна из сторон использует полудуплексный, а другая дуплексный режим передачи.

Следующий пример демонстрирует специфический случай рассогласования дуплекса. Интерфейс Gi0/2 коммутатора SW2 на рис. 10.4 был настроен командами `speed 100` и `duplex full` (отметим, что эти параметры не рекомендуются для интерфейса GigabitEthernet). Команды `speed` и `duplex` на коммутаторах Cisco отключают автопереговоры IEEE на соответствующем порте. Если интерфейс Gi0/1 коммутатора SW1 попытается использовать автопереговоры, то он также установит скорость 100 Мбит/с, но стандартно применит полудуплекс. Пример 10.4 демонстрирует результаты данного конкретного случая на коммутаторе SW1.

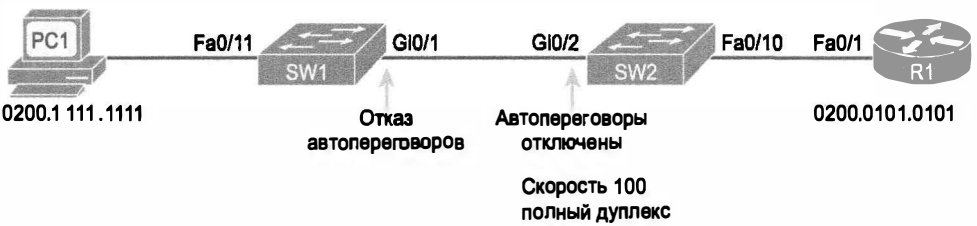


Рис. 10.4. Условия для рассогласования дуплекса между коммутаторами SW1 и SW2

Пример 10.4. Подтверждение рассогласования дуплекса на коммутаторе SW1

```
SW1# show interfaces gi0/1 status

Port  Name               Status       Vlan  Duplex  Speed  Type
Gi0/1                    connected    trunk a-half  a-100  10/100/1000BaseTX
```

Сначала рассмотрим вывод команды, подтверждающий скорость и дуплекс на коммутаторе SW1. Он также выводит префикс `a-`, указывающий на автопереговоры. Хотя коммутатор SW1 использовал стандартные значения, вывод команды отметит их как полученные в результате автопереговоров.

Рассмотренный выше пример иллюстрирует также одну сложную проблему рассогласования дуплекса интерфейсов: коммутатор SW1 работает в полудуплексном режиме, а SW2 — в дуплексном. Обнаружить рассогласование дуплекса крайне проблематично и намного сложнее, чем рассогласование скоростей, поскольку *при несовпадении настройки дуплекса на концах канала Ethernet интерфейс коммутатора все равно будет сообщать о его рабочем состоянии, т.е.* `connect` *(или* `up/up` *в другой команде)*. Такой интерфейс будет работать, но, скорее всего, очень плохо, производительность его будет невысока, и для него будут характерны перемежающиеся отказы и проблемы. Причиной нестабильной работы интерфейса является использование в полудуплексном режиме работы логики *множественного доступа с контролем несущей и обнаружением коллизий* (Carrier Sense Multiple Access with Collision Detection — CSMA/CD), которая ожидает окончания приема фрейма, чтобы начать собственную передачу, и предполагает, что в противном случае возникнет коллизия. В действительности коллизии в таком варианте работы соединения физически не возникнет, но коммутатор-то этого не знает! Вторая проблема в том, что при наличии интенсивного трафика интерфейс коммутатора будет сигнализировать о нормальном подключенном состоянии, но трафик через него не будет передаваться или будет передаваться нестабильно.

Чтобы обнаружить несоответствие режимов дуплексности на концах канала, следует проверить настройки на обоих интерфейсах, посмотреть, увеличиваются ли счетчики *коллизий* (collision) и *запоздалых коллизий* (late collision), как показано в следующем разделе.

Проблемы уровня 1 работающих интерфейсов

Когда интерфейс находится в состоянии `connected` (или `up/up`), коммутатор полагает, что он работает. Коммутатор, конечно, пытается использовать интерфейс, но в то же время он хранит разные счетчики интерфейса. Счетчики помогают выявить проблемы, казалось бы, работающего интерфейса. В данном разделе рассматриваются некоторые из сопутствующих концепций и несколько наиболее распространенных проблем.

При любых физических проблемах передачи принимающее устройство может получить фрейм, биты которого изменили значения. Такие фреймы не пройдут проверку системы обнаружения ошибок, реализованной на базе поля FCS в концевике Ethernet (см. главу 2). Принимающее устройство отбрасывает фрейм и относит его на счет некой *ошибки ввода* (input error). Коммутаторы Cisco отображают эту ошибку как ошибку CRC, см. пример 10.5. (Термин *циклический избыточный код*

(Cyclic Redundancy Check — CRC) связан со способом работы математического механизма обнаружения ошибок FCS.)

Пример 10.5. Счетчики интерфейса для определения проблем уровня 1

```
SW1# show interfaces fa0/13
! Часть выводимых строк опущена для краткости
    Received 284 broadcasts (0 multicast)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 281 multicast, 0 pause input
    0 input packets with dribble condition detected
    95226 packets output, 10849674 bytes, 0 underruns
    0 output errors, 0 collisions, 1 interface resets
    0 unknown protocol drops
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier, 0 PAUSE output
    0 output buffer failures, 0 output buffers swapped out
```

Количество ошибок ввода и ошибок CRC — это лишь некоторые из счетчиков в выводе команды `show interfaces`. Довольно не просто решить, какие счетчики позволяют выявить существующую проблему, а о каких можно не беспокоиться.

В примере выделено несколько счетчиков для демонстрации, с чего можно начать поиск счетчиков, свидетельствующих о проблеме, и как отличить их от отображающих нормальные события. Ниже приведен список каждого выделенного счетчика с их кратким описанием (в порядке расположения в примере).

`runts` (карлики). Фреймы, не отвечающие минимальному требованию к размеру фрейма (64 байта, включая 18 байтов для полей MAC-адреса получателя, MAC-адреса отправителя, типа и FCS). Могут быть вызваны коллизиями.

`giants` (гиганты). Фреймы, размер которых превышает максимально допустимый (1518 байтов, включая 8 байтов для полей MAC-адреса получателя, MAC-адреса отправителя, типа и FCS).

`input errors` (ошибки ввода). Сумма значений нескольких счетчиков, включая `including runts, giants, no buffer, CRC, frame, overrun` и `ignored`.

`CRC`. Полученные фреймы, не прошедшие проверку FCS; могут быть вызваны коллизиями.

`frame` (фрейм). Полученные фреймы с неверным форматом (например, завершающиеся неполным байтом); могут быть вызваны коллизиями.

`packets output` (вывод пакетов). Общее количество пакетов (фреймов), покинувших интерфейс.

`output errors` (ошибки вывода). Общее количество пакетов (фреймов), которые порт коммутатора пытался передавать, но столкнулся с проблемой.

`collisions` (коллизии). Счетчик всех коллизий, произошедших при передаче фреймов интерфейсом.

`late collisions` (запоздалые коллизии). Подмножество всех коллизий, произошедших после передачи 64-го байта фрейма. (В правильно работающей локальной сети Ethernet коллизии происходят в пределах первых 64 байтов; запоздалые коллизии зачастую указывают на рассогласование дуплекса.)

Обратите внимание, что большинство этих счетчиков участвует в процессе CSMA/CD, когда используется полудуплекс. Коллизии являются нормальным явлением при полудуплексном режиме передачи с использованием процесса CSMA/CD, поэтому у интерфейса коммутатора с увеличенным счетчиком коллизий даже может и не быть проблемы. Однако запоздалые коллизии указывают на классическую проблему рассогласования дуплекса.

Если в процессе разработки и развертывания сети были соблюдены стандарты, то все коллизии должны появляться при передаче заголовка фрейма, т.е. до конца 64-го байта. Если же коммутатор уже успел передать 64 байта фрейма и после этого обнаружил коллизию, то такая коллизия называется *запоздалой* (late collision) и устройство увеличивает именно счетчик запоздалых коллизий для интерфейса. Помимо этого, коммутатор посылает *сигнал оповещения о коллизии* (jamming signal), как предусмотрено алгоритмом CSMA/CD, приостанавливает передачу на случайный период времени и только по его истечении пробует повторно передать данные.

При рассогласовании дуплекса, как у коммутаторов SW1 и SW2 на рис. 10.4, на полудуплексном интерфейсе, вероятно, будет наблюдаться увеличение счетчика запоздалых коллизий. Почему? Полудуплексный интерфейс (коммутатора SW1) посылает фрейм, но дуплексный интерфейс соседнего устройства (SW2) не ждет окончания передачи и посылает фрейм в любой момент, даже после 64-го байта фрейма, посланного полудуплексным коммутатором. Поэтому достаточно повторить команду `show interfaces` несколько раз, и если значение счетчика запоздалых коллизий увеличивается на полудуплексном интерфейсе, то причина проблемы в рассогласовании дуплекса.

Рабочий интерфейс (в состоянии `up/up`) может также страдать от проблем, связанных с физической кабельной проводкой. Повреждение может быть недостаточно серьезным для полного прекращения передачи, но некоторые фреймы проходят по кабелю уже с ошибками. Например, чрезмерные помехи на кабеле способны увеличить значение различных счетчиков ошибок ввода, особенно счетчика CRC. В частности, если растет количество ошибок CRC, а счетчика коллизий нет, то причиной проблем могут быть просто помехи в кабеле. (Коммутатор считает каждую коллизию фрейма одной из форм ошибки ввода.)

Прогноз перенаправления фреймов коммутаторами

Этот, четвертый из пяти главных разделов данной главы начинается с рассмотрения ключевой части процесса поиска неисправностей в локальных сетях Ethernet: прогноза пути следования фреймов по локальной сети и сравнения ожидаемого с фактически происходящим.

Прогноз содержимого таблицы MAC-адресов

Как уже упоминалось в главе 6, коммутаторы обнаруживают MAC-адреса отправителей во входящих фреймах и заносят их в специализированные таблицы, чтобы в перспективе выполнять перенаправление и фильтрацию для всех входящих в устройство фреймов. Чтобы представить себе, как именно данный коммутатор будет обрабатывать фрейм Ethernet и на какой интерфейс он его перенаправит, следует научиться просматривать и интерпретировать таблицу MAC-адресов коммутаторов компании Cisco.

Команда `show mac address-table` выводит содержимое таблицы MAC-адресов коммутатора. Вывод этой команды содержит как некоторые дополнительные статические адреса, а именно MAC-адреса коммутатора, статически заданные адреса, включая указанные защитой портов, так и динамически изученные устройством. Если нужно просмотреть только динамические записи MAC-адресов в таблице коммутации, следует использовать команду с дополнительным параметром `show mac address-table dynamic`.

ВНИМАНИЕ!

Некоторые устаревшие коммутаторы поддерживают только прежнюю версию этой команды: `show mac-address-table`.

Формально процесс поиска неисправности начинается с прогноза путей следования фреймов по локальной сети. Вернемся к примеру и рис. 10.3 и попробуем составить на бумаге таблицу MAC-адресов для каждого коммутатора. Включите в таблицу MAC-адреса обоих компьютеров, а также MAC-адрес Gi0/1 маршрутизатора R1. (Подразумевается, что все они находятся в сети VLAN 10.) Теперь попробуем составить прогноз относительно того, какие интерфейсы будут использоваться для перенаправления фрейма, посланного компьютерами Фреда, Барни и маршрутизатором R1 на любое устройство.

Предполагаемый путь следования фреймов определяют в данном случае составленные записи таблицы MAC-адресов. Даже при том, что пример сети на рис. 10.3 имеет только один физический путь через сеть Ethernet, упражнение имеет смысл, поскольку позволяет увязать то, что ожидалось в таблице MAC-адресов, с тем, как коммутаторы перенаправляют фреймы. На рис. 10.5 приведен результат создания записей таблицы MAC-адресов для компьютеров Фреда, Барни и маршрутизатора R1.

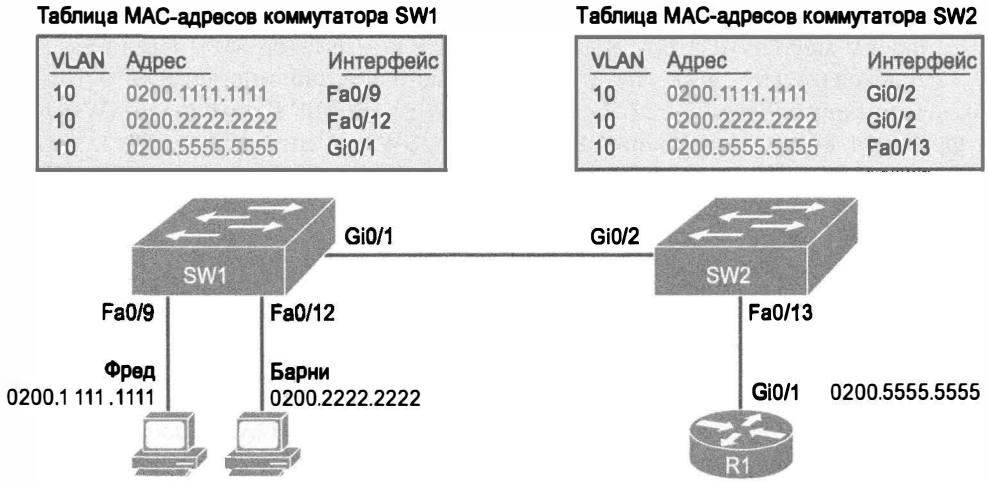


Рис. 10.5. Прогноз записей в таблице MAC-адресов на коммутаторах SW1 и SW2

В примере 10.6 отображается фактическое представление концепций, приведенных на рис. 10.5, в виде вывода команды `show mac address-table dynamic` на обоих коммутаторах. Команды отображают все динамически изученные записи таблицы MAC-адресов на коммутаторе для всех сетей VLAN.

Пример 10.6. Исследование динамических записей таблиц MAC-адресов коммутаторов SW1 и SW2

```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan Mac Address      Type      Ports
---- --------------- --------- -----
10   0200.1111.1111  DYNAMIC   Fa0/9
10   0200.2222.2222  DYNAMIC   Fa0/12
10   0200.5555.5555  DYNAMIC   Gi0/1


SW2# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan Mac Address      Type      Ports
---- --------------- --------- -----
10   0200.1111.1111  DYNAMIC   Gi0/2
10   0200.2222.2222  DYNAMIC   Gi0/2
10   0200.5555.5555  DYNAMIC   Fa0/13
```

В процессе прогнозирования MAC-адресов, которые будут присутствовать в таблицах коммутаторов, следует представить себе, как именно будет передаваться фрейм с одного конца локальной сети на другой, а после этого записать, через какие порты коммутатора он будет проходить. Например, когда компьютер Барни пересылает фрейм маршрутизатору R1, он попадет в коммутатор SW1 через интерфейс Fa0/12, следовательно, у коммутатора будет запись в таблице адресов о том, что MAC-адрес Барни 0200.2222.2222 связан с портом Fa0/12. Коммутатор SW1 отправит фрейм от компьютера Барни коммутатору SW2 на интерфейс Gi0/2, поэтому в таблице адресов SW2 также будет присутствовать MAC-адрес компьютера Барни (0200.2222.2222), и он будет связан с портом Gi0/2.

После того как содержимое таблиц адресов было спрогнозировано, следует проверить, что в действительности происходит на коммутаторах, — об этом и пойдет речь в следующем разделе.

Анализ маршрута фреймов

Процесс поиска и устранения неисправностей основан на трех основных идеях: прогнозе того, что должно происходить, определении происходящего фактически и выяснении различий между прогнозом и фактом. В следующем разделе обсуждается фактически происходящее в сети VLAN на основании таблицы MAC-адресов: сначала на основании логики перенаправления коммутатора, а затем на конкретном примере.

В следующем списке суммируется логика перенаправления коммутатора, включая средства коммутации локальной сети, обсуждаемые в этой книге.

Этапы передачи фреймов коммутаторами

Ключевая тема

Этап 1 Если входящий интерфейс в настоящее время находится в состоянии `up/up` (`connected`), процесс происходит следующим образом:

A. Если порт защищен, выполняется логика фильтрации фреймов.

B. Если это порт доступа, определяется доступная интерфейсу сеть VLAN.

C. Если это порт магистрального канала, определяется сеть VLAN для данного фрейма

Этап 2 Принимается решение о перенаправлении. В таблице MAC-адресов осуществляется поиск MAC-адреса получателя фрейма, но только среди тех записей, которые выявлены на этапе 1. Если MAC-адрес получателя...

A. *Найден (одноадресатный)*, фрейм перенаправляется на единственный интерфейс, указанный в соответствующей записи таблицы адресов.

B. *Не найден (одноадресатный)*, фрейм лавинно рассылается на все другие порты доступа (кроме входящего) в той же сети VLAN, а также на магистральные каналы (как обсуждалось в главе 9, согласно команде `show interfaces trunk`).

C. *Широковещательный* фрейм лавинно рассылается по тем же правилам, что и в предыдущем случае

Рассмотрим этот процесса на примере передачи фрейма компьютером Барни на его стандартный шлюз R1 (0200.5555.5555). Согласно этим этапам, происходит следующее.

Этап 1 Процесс входного интерфейса:

A. Защита порта отсутствует.

B. Коммутатор SW1 получает фрейм на интерфейс Fa0/12, порт доступа сети VLAN 10

Этап 2 Принимается решение о перенаправлении: коммутатор SW1 просматривает в своей таблице MAC-адресов записи для сети VLAN 10:

A. Коммутатор SW1 находит запись для известного одноадресатного адреса 0200.5555.5555, связанного с сетью VLAN 10, где указан исходящий интерфейс Gi0/1. Таким образом, коммутатор SW1 перенаправляют фрейм только на интерфейс Gi0/1. (Это магистральный канал связи VLAN, поэтому коммутатор SW1 добавляет тег сети VLAN 10 в заголовок 802.1Q магистрального канала.)

Теперь фрейм с адресом отправителя 0200.2222.2222 (Барни) и адресом получателя 0200.5555.5555 (R1) поступает на маршрутизатор SW2. Применим теперь ту же логику к маршрутизатору SW2.

Этап 1 Процесс входного интерфейса:

A. Защита порта отсутствует.

B. Коммутатор SW2 получает фрейм на интерфейс Gi0/2 магистрального канала; фрейм содержит тег сети VLAN 10. (Коммутатор SW2 удалит заголовок 802.1Q.)

Этап 2 Принимается решение о перенаправлении: коммутатор SW2 просматривает в своей таблице MAC-адресов записи для сети VLAN 10:

A. Коммутатор SW2 находит запись для известного одноадресатного адреса 0200.5555.5555, связанного с сетью VLAN 10, где указан исходящий интерфейс Fa0/13. Таким образом, коммутатор SW2 перенаправляют фрейм только на интерфейс Fa0/13

Теперь фрейм проследует по кабелю Ethernet от коммутатора SW2 на маршрутизатор R1.

Защита порта и фильтрация фреймов

При следовании фреймов через коммутаторы LAN на их пути могут встретиться различные виды фильтров, способных отбросить фреймы, даже когда интерфейс работает. Например, коммутаторы LAN могут использовать такие фильтры, как *списки управления доступом* (Access Control List — ACL), фильтрующие фреймы на основании MAC-адресов отправителя и получателя. Кроме того, маршрутизаторы могут фильтровать пакеты IP, используя списки ACL IP. (В настоящей книге не рассматриваются списки ACL коммутаторов LAN, а списки ACL IP маршрутизаторов обсуждаются в главе 22.)

Кроме того, защита порта (см. главу 8) также фильтрует фреймы. В некоторых случаях срабатывание защиты порта определяется легко, поскольку интерфейс оказывается отключенным. Однако в других случаях защита порта оставляет интерфейс включенным, но он просто отбрасывает подозрительный трафик. С точки зрения поиска и устранения неисправностей конфигурация защиты порта, при которой интерфейс остается включенным, но отбрасывает фреймы, требует от сетевого инженера внимательно изучить состояние защиты порта, а не просто просмотреть интерфейсы и таблицу MAC-адресов.

Напомним, что защита порта допускает три реакции на нарушение (`shutdown`, `protect` и `restrict`), но только стандартная настройка, `shutdown`, заставляет коммутатор отключать интерфейс.

Для примера рассмотрим случай, когда в рабочей сети некто применил защиту порта, чтобы фильтровать фреймы, посланные Барни. Используемая топология сети приведена на рис. 10.3 и 10.5. Компьютер Барни посылает фреймы на порт Fa0/12 коммутатора SW1, на котором теперь установлена защита. Конфигурация защиты порта рассматривает фреймы с MAC-адресом отправителя Барни как недопустимые и использует для них действие `protect`.

Что произойдет? Теперь коммутатор SW1 отбросит всех фреймы с MAC-адресом Барни. Но он не отключит интерфейс. Команды `show interfaces` и `show interfaces status` на коммутаторе SW1 не покажут изменений в состоянии интерфейса и не дадут никаких доказательств происходящему. Чтобы подтвердить отбрасывание посланных Барни фреймов защитой порта, придется просмотреть конфигурацию защиту порта (команда `show port-security interface`).

Таблица MAC-адресов дает некоторую подсказку о применении защиты порта. Поскольку защита порта управляет MAC-адресами, все связанные с таким портом MAC-адреса отображаются как статические. В результате команда `show mac address-table dynamic` не отобразит MAC-адреса тех интерфейсов, на которых включена защита порта. Однако команды `show mac address-table` и `show mac address-table static` отобразят эти MAC-адреса как статические.

Анализ сетей VLAN и магистральных каналов VLAN

Как уже упоминалось, процесс перенаправления коммутатора частично зависит от сети VLAN и магистрального соединения VLAN. Прежде чем коммутатор сможет перенаправлять фреймы в определенную сеть VLAN, коммутатор должен знать о ее существовании и она должна быть рабочей. Кроме того, коммутатор сможет перенаправлять фреймы по магистральному каналу VLAN, только если это разрешено.

Этот заключительный раздел посвящен проблемам сетей VLAN и магистральных каналов VLAN, а именно их влиянию на процесс коммутации фреймов. Ниже приведены четыре потенциальных проблемы и способы их решения.

Этап 1 Выявите все интерфейсы доступа и связанные с ними сети VLAN. По мере необходимости переназначьте интерфейсы правильным сетям VLAN

Этап 2 Выясните, существует ли сеть VLAN на каждом коммутаторе и является ли она активной (настроена или выявлена протоколом VTP). Если нет, то настройте и активизируйте сеть VLAN

Этап 3 Проверьте списки разрешенных сетей VLAN на коммутаторах с обоих концов магистрального канала и удостоверьтесь в их совпадении

Этап 4 Удостоверьтесь, что для любых каналов связи, которые должны использовать магистральное соединение, один коммутатор не рассматривает его как магистральное соединение, а другой коммутатор рассматривает. Проверьте это магистральное соединение на предмет неудачного выбора параметров конфигурации

Проверка принадлежности интерфейсов доступа к соответствующим сетям VLAN

Следует удостовериться, что каждому интерфейсу доступа была присвоена правильная сеть VLAN. Инженер должен определить, какие интерфейсы коммутатора являются портами доступа, а какие магистральных каналов, определить сети VLAN на каждом интерфейсе доступа и сравнить полученную информацию с документацией. При этом перечисленные в табл. 10.4 команды `show` могут быть особенно полезны.

Таблица 10.4. Команды, позволяющие найти порты доступа и сети VLAN

Ключевая тема

Пользовательские команды	Описание
`show vlan brief` `show vlan`	Выводит каждую сеть VLAN и все присвоенные ей интерфейсы (но не включая работающие магистральные каналы)
`show vlan id` *номер*	Выводит порты доступа и магистрального канала сети VLAN
`show interfaces` *тип номер* `switchport`	Выводит сеть VLAN интерфейса доступа, голосовую сеть VLAN, заданный и рабочий режимы (доступа или магистрального канала)
`show mac address-table`	Выводит записи таблицы MAC-адресов, включая ассоциированные сети VLAN

Если возможно, этот этап стоит начать с команды `show vlan` или `show vlan brief`, поскольку они перечисляют все известные сети VLAN и интерфейсы доступа, принадлежащие каждой из них. Однако они не отображают работающие магистральные каналы. В выводе будут перечислены все интерфейсы (кроме участвующих в настоящий момент в магистральном соединении), независимо от того, находятся ли они в рабочем или нерабочем состоянии.

Если в экзаменационном вопросе команды `show vlan` и `show interface switchport` не доступны, выявить интерфейсы доступа сети VLAN поможет также команда `show mac address-table`. Она выводит таблицу MAC-адресов, каждая запись которой включает MAC-адрес, интерфейс и идентификатор VLAN. Если экзаменационный вопрос подразумевает, что интерфейс коммутатора соединен с одиночным устройством (компьютером), достаточно выявить только одну запись таблицы MAC-адресов, в которой указан данный конкретный интерфейс доступа; указанный

в этой записи идентификатор VLAN идентифицирует сеть доступа VLAN. (Для магистральных интерфейсов таких предположений сделать нельзя.)

Определив интерфейсы доступа и связанные с ними сети VLAN, можно выявить несоответствия и при необходимости использовать подкоманду интерфейса `switchport access vlan` *идентификатор_vlan*, чтобы назначить правильные идентификаторы VLAN.

Доступ. VLAN не определена

Коммутаторы не перенаправляют фреймы для тех сетей VLAN, которые либо не настроены, либо настроены, но отключены (состояние `shutdown`). В данном разделе рассмотрены наилучшие способы, позволяющие подтвердить, что коммутатору известно о существовании конкретной сети VLAN и что она находится в рабочем состоянии.

Конкретная сеть VLAN может быть определена на коммутаторе двумя способами: глобальной командой конфигурации `vlan` *номер* или при помощи протокола VTP с другого коммутатора. В этой книге намеренно игнорируется протокол VTP в максимально возможной степени, поэтому будем полагать, что единственный способ определить сеть VLAN на коммутаторе — это применение команды `vlan` на локальном коммутаторе.

Далее, команда `show vlan` всегда выводит все известные коммутатору сети VLAN, а команда `show running-config` — нет. Коммутаторы, настроенные как серверы и клиенты VTP, не отображают команды `vlan` ни в конфигурационном файле `running-config`, ни в файле `startup-config`; на этих коммутаторах нельзя использовать команду `show vlan`. Коммутаторы с настроенным прозрачным режимом VTP или отключенным протоколом VTP, напротив, выводят команды `vlan` в файлах конфигурации. (Чтобы узнать текущий режим протокола VTP на коммутаторе, используйте команду `show vtp status`.)

Если причина проблемы в том, что сеть VLAN не существует, ее просто следует создать (процесс подробно описан в главе 9).

Доступ. VLAN отключена

Если сеть VLAN существует, имеет смысл проверить, активна ли она. Команда `show vlan` должна указать одно из двух состояний сети VLAN: `active` (активна) или `act/lshut`. Второе состояние означает, что сеть VLAN отключена. Отключение сети VLAN на коммутаторе означает только то, что он *не перенаправляет фреймы этой сети VLAN*.

Коммутатор IOS предоставляет два похожих метода конфигурации, позволяющих отключить (`shutdown`) и включить (`no shutdown`) сеть VLAN. Пример 10.7 демонстрирует сначала использование глобальной команды `[no] shutdown vlan` *номер*, а затем подкоманды режима VLAN `[no] shutdown`. Пример демонстрирует глобальные команды включения и отключения сетей VLAN 10 и 20 соответственно, а также использование подкоманд VLAN для включения и отключения сетей VLAN 30 и 40 соответственно.

Пример 10.7. Включение и отключение сети VLAN на коммутаторе

```
SW2# show vlan brief

VLAN Name                    Status     Ports
```

```
---- -------------------- ---------- ---------------------------------
1    default              active     Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                     Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                     Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                     Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                     Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                     Fa0/22, Fa0/23, Fa0/24, Gi0/1
10   VLAN0010             act/lshut  Fa0/13
20   VLAN0020             active
30   VLAN0030             act/lshut
40   VLAN0040             active
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# no shutdown vlan 10
SW2(config)# shutdown vlan 20
SW2(config)# vlan 30
SW2(config-vlan)# no shutdown
SW2(config-vlan)# vlan 40
SW2(config-vlan)# shutdown
SW2(config-vlan)#
```

Проверка списка разрешенных сетей VLAN на обоих концах магистрального канала

Два предыдущих случая присущи плохому выбору конфигурации на магистральном канале VLAN. В реальной жизни достаточно правильно настроить магистральный канал, как описано в главе 9 и разделе, следующим за этим. Но на экзаменах следует быть готовым к некоторым неожиданностям, возможным при неудачном выборе конфигурации магистральных каналов.

На противоположных концах магистрального канала VLAN могут быть настроены разные списки разрешенных сетей VLAN. При их несовпадении магистральный канал не сможет передать трафик некоторых сетей VLAN. Пример приведен на рис. 10.6. На обоих коммутаторах определены сети VLAN 1–10, поэтому оба стандартно включают их в свои списки разрешенных сетей VLAN. Однако коммутатор SW2 был перенастроен командой `switchport trunk allowed vlan remove 10`, удалившей сеть VLAN 10 из списка разрешенных сетей порта G0/2 коммутатора SW2. В данном случае на коммутаторе SW1 сеть VLAN 10 все еще разрешена и нормально работает, отмечая и перенаправляя фреймы в сеть VLAN 10 (этап 1 на рисунке), но коммутатор SW2 просто отбрасывает все фреймы для сети VLAN 10, получаемые на этом магистральном канале (этап 2), поскольку трафик сети VLAN 10 на этом магистральном канале коммутатор SW2 уже не считает разрешенным.

И самое интересное: эту проблему нельзя обнаружить только с одной или только с другой стороны магистрального канала. Вывод команды `show interfaces trunk` с обеих сторон выглядит совершенно нормально. Проблему можно заметить, только сравнивая списки разрешенных сетей на обоих концах магистрального канала.

Чтобы сравнить списки, необходимо просмотреть вторые из трех списков сетей VLAN, выводимых командой `show interfaces trunk`, как выделено в выводе примера 10.8. Выделенный текст демонстрирует второй раздел, списки VLAN которого соответствуют таким критериям: сети VLAN, существующие на коммутаторе, не отключенные и не удаленные из списка разрешенных.

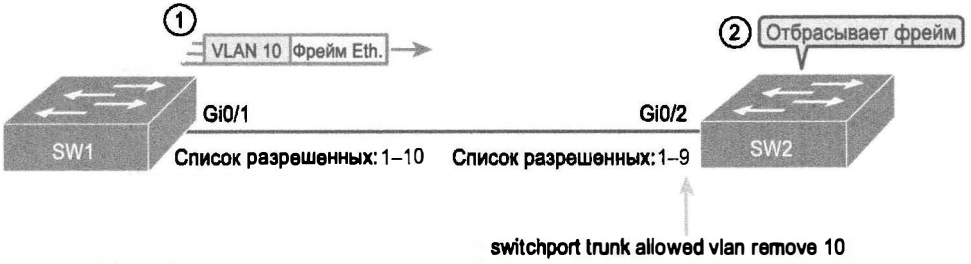


Рис. 10.6. Несовпадение списков разрешенных сетей VLAN на магистральном канале

Пример 10.8. Второй набор сетей VLAN: существующие, не отключенные и разрешенные

```
SW2# show interfaces trunk

Port     Mode          Encapsulation  Status     Native vlan
Gi0/2    desirable 802.1q             trunking   1

Port     Vlans allowed on trunk
Gi0/2    1-4094

Port     Vlans allowed and active in management domain
Gi0/2    1-9

Port     Vlans in spanning tree forwarding state and not pruned
Gi0/2    1-9
```

Рассогласование рабочего состояния магистрали

Магистральное соединение может быть настроено правильно, чтобы оба коммутатора перенаправили фреймы для одинакового набора сетей VLAN. Но магистральные каналы могут быть и рассогласованными по нескольким разным причинам. В одних случаях оба коммутатора полагают, что их интерфейсы не формируют магистральный канал. В других случаях один коммутатор полагает, что его интерфейс принадлежит магистральному соединению, а другой коммутатор — нет.

Наиболее распространенная ошибка конфигурации (когда оба коммутатора не считают канал магистральным) возникает при вводе команды `switchport mode dynamic auto` на обоих коммутаторах канала связи. Слово `auto` лишь заставляет всех полагать, что канал связи автоматически станет магистральным, но на самом деле она переводит оба порта в режим пассивного ожидания. В результате оба коммутатора ждут начала переговоров от устройства на противоположном конце канала связи.

При такой специфической ошибке конфигурации команда `show interfaces switchport` на обоих коммутаторах подтверждает административное состояние (`auto`), а также тот факт, что оба коммутатора работают как статические порты доступа (`static access`). Это выделено в примере 10.9 вывода данной команды.

Пример 10.9. Рабочее состояние магистрали

```
SW2# show interfaces gigabit0/2 switchport
Name: Gi0/2
Switchport: Enabled
```

```
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dotlq
Operational Trunking Encapsulation: native
! строки опущены для краткости
```

В результате другой ошибки конфигурации магистрали один коммутатор находится в рабочем состоянии `trunk` (магистраль), а другой в рабочем состоянии `static access` (статический доступ). При такой комбинации интерфейс работает плохо. Состояние на каждом конце будет `up/up` или `connected`. Фактически трафик собственной сети VLAN будет пересекать канал связи успешно, а трафик всех остальных сетей VLAN — нет.

На рис. 10.7 приведена неправильная конфигурация с магистралью на одном конце и без нее на другом. Сторона с магистральным каналом (в данном случае SW1) разрешает магистральное соединение (команда `switchport mode trunk`). Но эта команда не отключает переговоры DTP. В связи с этой специфической проблемой коммутатор SW1 также отключает переговоры DTP, используя команду `switchport nonegotiate`. Конфигурация коммутатора SW2 также позволяет создать проблему при использовании параметров магистрали, полагающихся на переговоры DTP. Поскольку коммутатор SW1 отключил переговоры DTP, таковые на коммутаторе SW2 терпят неудачу и магистральный канал он не создает.

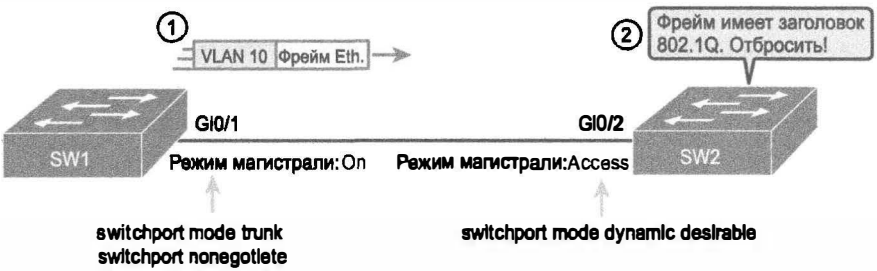


Рис. 10.7. Рассогласование рабочего состояния магистрали

В данном случае коммутатор SW1 рассматривает свой интерфейс G0/1 как магистральный, а коммутатор SW2 рассматривает свой интерфейс G0/2 как порт доступа (не магистральный). Как показано на этапе 1 рисунка, коммутатор SW1 мог бы, например, перенаправить фрейм сети VLAN 10 (этап 1). Но коммутатор SW2 рассматривает любой поступающий фрейм с заголовком 802.1Q как запрещенный, поскольку считает свой порт G0/2 портом доступа. Поэтому он отбрасывает на этом порту все фреймы с заголовком 802.1Q.

Встретившись с возможностью подобной проблемы, в первую очередь проверяйте рабочее состояние магистрального канала на обоих концах магистрали. Для выявления фактов, связанных с магистральным соединением, лучше всего подходят команды `show interfaces trunk` и `show interfaces switchport`.

ВНИМАНИЕ!

Откровенно говоря, в реальной жизни следует избегать конфигураций подобного вида. Но коммутаторы никак не препятствуют совершению таких ошибок, поэтому к ним следует быть готовым.

Обзор

Резюме

- Собственный протокол обнаружения устройств Cisco (CDP) позволяет получить базовую информацию о соседних маршрутизаторах и коммутаторах, даже не зная пароля для доступа к ним.
- Чтобы получить информацию, маршрутизаторы и коммутаторы рассылают сообщения CDP через все свои работающие интерфейсы. Такие сообщения содержат информацию об устройстве, которое его отправило; о других устройствах маршрутизатор или коммутатор узнает из принимаемых им аналогичных сообщений протокола CDP.
- С точки зрения процесса поиска и устранения неисправностей в сетях протокол CDP может использоваться для подтверждения или уточнения сетевой документации и схем, а также для обнаружения новых устройств и интерфейсов в сети.
- Протокол CDP позволяет выяснить несколько полезных фактов о соседних устройствах Cisco.
  - *Идентификатор устройства.* Обычно это название устройства.
  - *Список адресов*. Представляет собой адрес сетевого и канального уровней.
  - *Идентификатор порта*. Интерфейс дистанционного маршрутизатора или коммутатора на другом конце канала связи, пославший анонс CDP.
  - *Список возможностей.* Описывает тип устройства (например, соседнее устройство — это коммутатор или маршрутизатор).
  - *Платформа*. Показывает модель соседнего устройства.
- Команды `show interfaces` и `show interfaces description` коммутатора выдают двухкомпонентные коды состояний интерфейса точно так же, как и в маршрутизаторах. Такие коды описывают *состояние линии* и *состояние протокола* интерфейса.
- Команда `show interfaces status` выводит для каждого интерфейса одну короткую строку с одним кодовым словом его состояния. Такой код состояния интерфейса имеет однозначную привязку к двухкомпонентному традиционному коду состояния порта.
- Любое состояние интерфейса, отличное от `connect` или `up/up`, означает, что коммутатор не будет передавать и принимать фреймы через него.
- Следующий список демонстрирует краткое описание каждого счетчика, отображаемого в выводе команды `show interfaces`.
  - `runts` (карлики). Фреймы, не отвечающие минимальному требованию к размеру фрейма (64 байта, включая 18 байтов для полей MAC-адреса получателя, MAC-адреса отправителя, типа и FCS). Могут быть вызваны коллизиями.

- `giants` (гиганты). Фреймы, размер которых превышает максимально допустимый (1518 байтов, включая 8 байтов для полей MAC-адреса получателя, MAC-адреса отправителя, типа и FCS).
- `input errors` (ошибки ввода). Сумма значений нескольких счетчиков, включая `including runts, giants, no buffer, CRC, frame, overrun` и `ignored`.
- `CRC`. Полученные фреймы, не прошедшие проверку FCS; могут быть вызваны коллизиями.
- `frame` (фрейм). Полученные фреймы с неверным форматом, например завершающиеся неполным байтом; могут быть вызваны коллизиями.
- `packets output` (вывод пакетов). Общее количество пакетов (фреймов), покинувших интерфейс.
- `output errors` (ошибки вывода). Общее количество пакетов (фреймов), которые порт коммутатора пытался передавать, но столкнулся с проблемой.
- `collisions` (коллизии). Счетчик всех коллизий, произошедших при передаче фреймов интерфейсом.
- `late collisions` (запоздалые коллизии). Подмножество всех коллизий, произошедших после передачи 64-го байта фрейма. (В правильно работающей локальной сети Ethernet коллизии происходят в пределах первых 64 байтов; запоздалые коллизии зачастую указывают на рассогласование дуплекса).

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Представьте себе коммутатор, подключенный кабелем Ethernet к маршрутизатору; маршрутизатору в конфигурации присвоено имя *Hannah*. С помощью какой команды можно узнать версию операционной системы устройства `Hannah`, не устанавливая с ним сеанс Telnet? (Выберите два ответа.)

   А) `show neighbors Hannah`

   Б) `show cdp`

   В) `show cdp neighbors`

   Г) `show cdp neighbors Hannah`

   Д) `show cdp entry Hannah`

   Е) `show cdp neighbors detail`

2. Коммутатор подключен к маршрутизатору с именем хоста `Hannah`. С помощью какой команды можно установить модель маршрутизатора `Hannah`? (Выберите два ответа.)

   А) `show neighbors`

   Б) `show neighbors Hannah`

В) `show cdp`

Г) `show cdp interface`

Д) `show cdp neighbors`

Е) `show cdp entry Hannah`

3. Вывод команды `show interfaces status` для коммутатора модели 2960 показывает в строке состояния слово "`disabled`" (отключен) для интерфейса Fa0/1. Какое из указанных ниже утверждений справедливо для такого интерфейса? (Выберите три ответа.)

А) В конфигурации интерфейса указана команда `shutdown`.

Б) В выводе команды `show interfaces fa0/1` будут отображаться два кода состояний: административно выключен (`administratively down`) и выключен (`down`).

В) В выводе команды `show interfaces fa0/1` будут отображаться два кода состояний: включен (`up`) и выключен (`down`).

Г) Интерфейс в данном случае не будет пересылать фреймы.

Д) Интерфейс в данном случае будет пересылать фреймы.

4. Гигабитовый интерфейс 0/1 (Gi0/1) коммутатора SW1 подключен к гигабитовому интерфейсу 0/2 (Gi0/2) коммутатора SW2. Для интерфейса Gi0/2 коммутатора SW2 введены команды — `speed 1000` и `duplex full`. В коммутаторе SW1 используются стандартные настройки для порта Gi0/1. Какое из приведенных ниже утверждений справедливо для такого соединения между интерфейсами? (Выберите два ответа.)

А) Соединение между устройствами будет работать на скорости 1000 Мбит/с (т.е. 1 Гбит/с).

Б) Коммутатор SW1 будет пытаться установить скорость 10 Мбит/с, поскольку у коммутатора SW2 отключены автопереговоры IEEE.

В) Соединение будет работать на скорости 1 Гбит/с, но коммутатор SW1 будет работать в полудуплексном режиме, а SW2 — в дуплексном.

Г) Оба коммутатора будут работать в дуплексном режиме.

С помощью команды `show interfaces fa0/1` был получен следующий результат:

```
Full-duplex, 100Mbps, media type is 10/100BaseTX
```

Что из перечисленного ниже справедливо для соответствующего интерфейса? (Выберите два ответа.)

А) Скорость была явно задана с помощью команды `speed 100` в подрежиме конфигурации интерфейса.

Б) Скорость, возможно, была задана с помощью команды `speed 100` в подрежиме конфигурации интерфейса.

В) Дуплексный режим был явно задан с помощью команды `duplex full` в подрежиме конфигурации интерфейса.

Г) Дуплексный режим, возможно, был задан с помощью команды `duplex full` в подрежиме конфигурации интерфейса.

5. Какая из указанных ниже команд отображает записи таблицы MAC-адресов, используемых защитой порта коммутатора? (Выберите два ответа.)

   А) `show mac address-table dynamic`

   Б) `show mac address-table`

   В) `show mac address-table static`

   Г) `show mac address-table port-security`.

6. На коммутаторе Cisco Catalyst введена команда `show mac address-table`. Какие из следующих ответов описывают выводимую ею информацию? (Выберите два ответа).

   А) MAC-адрес.

   Б) IP-адрес.

   В) Идентификатор VLAN.

   Г) Тип (широковещательный, многоадресатный, одноадресатный)

7. Коммутаторы SW1 и SW2 уровня 2 соединены каналом связи через порты G0/1 (SW1) и G0/2 (SW2). Сетевой инженер хочет использовать на этом канале связи магистральное соединение 802.1Q. Команда `show interfaces g0/1 switchport` на коммутаторе SW1 демонстрирует следующий вывод:

```
SW1# show interfaces gigabit0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
```

   Что из следующего истинно для порта G0/2 коммутатора SW2?

   А) Согласно команде `show interfaces switchport` рабочим состоянием должно быть `trunk`.

   Б) Согласно команде `show interfaces switchport` административным состоянием должно быть `trunk`.

   В) Коммутатор SW2 должен использовать команду конфигурации `switchport mode trunk` на интерфейсе G0/2, или канал связи не будет использован как магистральное соединение.

   Г) Коммутатор SW2 может использовать команду конфигурации `switchport mode dynamic auto` как одну из возможностей заставить канал связи использовать магистральное соединение.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 10.5.

Таблица 10.5. Ключевые темы главы 10

Элемент	Описание	Страница
Список	Информация, получаемая с помощью протокола CDP	319
Табл. 10.1	Варианты команды show cdp, используемые для получения информации о смежных устройствах	320
Табл. 10.3	Коды состояния интерфейсов коммутаторов локальных сетей	324
Прим. 10.3	Отображение настроек скорости и дуплексности интерфейсов коммутаторов	325
Список	Стандартные правила автопереговоров IEEE	327
Список	Этапы передачи фреймов коммутаторами	333
Табл. 10.4	Команды, позволяющие найти порты доступа и сети VLAN	335

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

сосед CDP (CDP neighbor), состояние "up и up", состояние "connected", состояние "error disabled", изоляция проблемы (problem isolation), первопричина (root cause), рассогласование дуплекса (duplex mismatch)

Таблицы команд

В табл. 10.6 и 10.7 приведен список команд этой главы, а также даны их краткие описания. В главах 8 и 9 были перечислены команды, также имеющие отношение к темам данной главы, тем не менее их следует искать в конце соответствующих глав.

Таблица 10.6. Команды конфигурации коммутаторов Catalyst 2960

Команда	Описание
`Shutdown` `no shutdown`	Команды подрежима конфигурации интерфейсов для выключения и включения интерфейса соответственно
`switchport port-security violation {protect | restrict | shutdown}`	Команда подрежима конфигурации интерфейса, указывающая, что необходимо предпринять, если фрейм с неразрешенным MAC-адресом поступил на интерфейс с включенной защитой
`cdp run` `no cdp run`	Команды режима глобального конфигурации для включения и выключения протокола CDP на всем устройстве
`cdp enable` `no cdp enable`	Команды подрежима конфигурации интерфейсов для включения и выключения протокола CDP на определенном интерфейсе

Окончание табл. 10.6

Команда	Описание
`speed {auto | 10 | 100 | 1000}`	Команда подрежима конфигурации интерфейса для установки скорости передачи данных вручную
`duplex {auto | full | half}`	Команда подрежима конфигурации интерфейса для установки режима дуплексности вручную

Таблица 10.7. Пользовательские команды главы 10

Команда	Описание
`show mac address-table [dynamic | static] [address` *аппаратный_адрес*`] [interface` *интерфейс*`] [vlan` *сеть_vlan*`]`	Отображает таблицу MAC-адресов устройства
`show port-security [interface` *интерфейс*`] [address]`	Отображает настройки защиты порта
`show cdp neighbors [`*тип номер*`]`	Выводит по одной строке краткой информации для каждого соседнего устройства или информацию об устройстве, доступном через указанный в команде интерфейс
`show cdp neighbors detail`	Выводит подробную информацию (около 15 строк) по каждому соседнему устройству
`show cdp entry` *имя_устройства*	Выводит ту же информацию, что и команда `show cdp neighbors detail`, но для одного указанного в параметре команды устройства
`show cdp`	Сообщает, включен ли протокол CDP глобально, и отображает стандартные таймеры рассылки обновлений и хранения информации
`show cdp interface [`*тип номер*`]`	Сообщает, включен ли протокол CDP на указанном интерфейсе, и отображает стандартные таймеры рассылки обновлений и хранения информации для этого интерфейса
`show cdp traffic`	Выводит статистику отправленных и полученных сообщений CDP
`show interfaces [`*тип номер*`]`	Выводит подробную информацию о состоянии интерфейса, настройках и его счетчиках
`show interfaces description`	Отображает по одной строке информации на интерфейс с двумя элементами состояния (подобно команде состояния `show interfaces`) и включает описание всего, что настроено на интерфейсах
`show interfaces [`*тип номер*`] status`	Выводит краткую информацию о состоянии и настройках интерфейса, в том числе скорость и дуплексный режим, а также сообщает о том, были ли параметры согласованы автоматически

Окончание табл. 10.7

Команда	Описание
`show interfaces [`*тип номер*`] switchport`	Выводит разнообразные конфигурационные параметры и текущее рабочее состояние, включая подробности магистрального соединения VLAN, сетей доступа и голосовых VLAN, а также собственных сетей VLAN
`show interfaces [`*тип номер*`] trunk`	Отображает информацию обо всех (или только перечисленных в команде) работающих в настоящее время магистральных каналах и сетях VLAN, поддерживаемых на этих магистральных каналах
`show vlan brief`, `show vlan`	Перечисляет все VLAN и все присвоенные ей интерфейсы, но не магистральные каналы
`show vlan id` *номер*	Отображает порты доступа и магистрального канала VLAN
`show vtp status`	Отображает текущее состояние VTP, включая текущий режим

Ответы на контрольные вопросы:

1 Д и Е. **2** Д и Е. **3** А, Б и Г. **4** А и Г. **5** Б и Г. **6** Б и В. **7** А и В. **8** Г.

Обзор части II

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части II

Задача	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей команд по категориям		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части книги, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения книги.

Ответы на вопросы

Ответьте на вопросы обзора этой части книги, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения книги.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если не все понятно, уделите время повторному изучению.

Создайте диаграмму связей команд по категориям

Часть II знакомит с интерфейсом командной строки Cisco (CLI) и довольно большим количеством команд конфигурации и пользовательских команд.

Для сдачи экзамена не обязательно запоминать каждую команду и каждый параметр. Но следует помнить все настраиваемые категории конфигурации на коммутаторах LAN и, по крайней мере, первое слово (или два) большинства команд. Следующие упражнения с диаграммами связей призваны помочь организовать эти команды в памяти.

Создайте диаграмму связей со следующими категориями команд из этой части книги:

консоль и VTY, протокол SSH, поддержка коммутаторами протокола IPv4, защита порта, сети VLAN, магистральные каналы VLAN, протокол CDP, другие команды администрирования коммутатора, другие подкоманды интерфейса.

По каждой категории обдумайте все команды конфигурации и все пользовательские команды (по большей части команды `show`). По каждой категории сгруппируйте команды конфигурации отдельно от пользовательских команд. На рис. Ч2.1 приведен пример разветвления команд CDP.

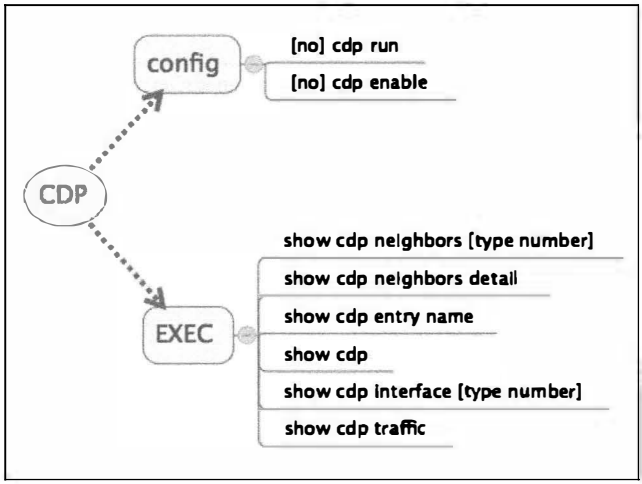


Рис. Ч2.1 Пример диаграммы связей ветви CDP

ВНИМАНИЕ!

Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения к данной книге.

И наконец, учтите следующие важные моменты при работе над этим проектом.

- Основная учебная задача этого упражнения осуществляется по мере его выполнения. Чтение других диаграмм связей или только таблиц команд не настолько способствует запоминанию необходимого материала.
- Выполните это задание, не подглядывая в книгу или свои записи.
- Завершив упражнение, сверьте результат с таблицами команд в конце глав и обратите внимание на первоначально забытые команды.
- Не волнуйтесь особенно о каждом пропущенном параметре или точности синтаксиса; достаточно записать лишь несколько первых слов команды.
- Делайте для последующего анализа примечания об абсолютно понятных командах и командах, в которых вы менее уверены.
- Повторите это упражнение впоследствии, когда появится десять свободных минут, и сравните результат с примечаниями, сделанными в прошлый раз.

Ответы приведены в приложении П (Appendix O) на веб-сайте, но ваши диаграммы связей могут выглядеть иначе.

Части III–VI посвящены средствам, тесно связанным с использованием протокола IP версии 4 (IPv4). Данная часть сосредоточивается на основных принципах IPv4-адресации и создания подсетей. В главе 11 приводится обзор IPv4-адресации на примере типичной корпоративной сети. В главах 12–14 рассматриваются некоторые специфические вопросы, связанные с использованием сети IPv4. Обратите внимание: в части V также обсуждаются подробности, связанные с IPv4-адресацией.

Часть III. IPv4-адресация и создание подсетей

ГЛАВА 11

Перспективы создания подсетей IPv4

Большинство простых задач по сетям требует, чтобы при работе и поиске неисправностей использовался уже существующий план подсетей и IP-адресации. Экзамены CCENT и CCNA оценивают готовность использовать уже существующую информацию об IP-адресации и подсетях для решения типичных задач, таких как контроль сети, реакция на возможные проблемы и их устранение.

Не только экзаменационные вопросы, но и проблемы реальных сетей требуют понимания их проекта. Процесс мониторинга какой-нибудь сети требует непрерывного ответа на вопрос: работает ли сеть так, как задумано? Если проблема есть, следует задаться таким вопросом: что происходит, когда сеть работает нормально и что сейчас не так? Оба вопроса требуют понимания проекта сети, включая подробности IP-адресации и подсетей.

В этой главе представлено несколько точек зрения и решений для серьезных проблем в IPv4-адресации. Какие адреса применять, чтобы они работали правильно? Когда сказано, что используются определенные номера, то что это говорит о выборе, сделанном неким другим сетевым инженером? Как сделанный выбор влияет на практическую задачу настройки коммутаторов, маршрутизаторов и хостов, использующих сеть регулярно? Данная глава пытается ответить на эти вопросы, раскрывая подробности работы IPv4-адресации.

В этой главе рассматриваются следующие экзаменационные темы
IP-адресация (IPv4/IPv6)
Работа и необходимость использования частных и открытых IP-адресов при IPv4-адресации.
Выбор подходящей схемы IPv4-адресации (использующей VLSM и суммирование), удовлетворяющей требования адресации в среде LAN/WAN.

Основные темы

Введение в подсети

Предположим, вам случилось попасть в магазин, когда там продавали самый длинный в мире бутерброд. Вы хотите есть, поэтому и зашли сюда. Теперь у вас есть один бутерброд, но длиной больше двух километров. Вы понимаете, что это немного больше, чем необходимо на обед. Чтобы сделать бутерброд более употребляемым (и переносимым), вы делите его на меньшие части и раздаете их другим людям вокруг вас, которые также не против пообедать.

На самом деле основная концепция создания подсетей очень похожа на случай с бутербродом. Вы начинаете с одной сети, но это только одна большая сеть. Как единый большой объект, она не очень полезна и слишком велика. Чтобы сделать ее более полезной, вы разделяете ее на меньшие части, называемые *подсетями* (subnet), а получив эти подсети, используете их в различных частях объединенной корпоративной сети.

Начинается этот раздел с краткого введения в создание подсетей IP. Вначале будут представлены общие концепции проектирования подсетей, когда одну сеть (или подсеть) действительно разделяют на подсети. Затем будут описаны многочисленные этапы, необходимые для создания подсети именно заданного проекта. К концу данного раздела у вас должно выработаться представление о том, что происходит на последующих этапах создания подсетей, обсуждаемых в остальной части данной главы.

ВНИМАНИЕ!

Эта глава и фактически все остальные главы данной книги вплоть до главы 25 посвящены протоколу IPv4, а не IPv6. Все упоминания о протоколе IP относятся к версии IPv4, если не указано иное.

Создание подсетей на простом примере

Сеть IP (IP network) — другими словами, сеть класса A, B или C — это просто набор последовательно пронумерованных IP-адресов, подчиняющихся неким предварительно установленным правилам. Правила классов A, B и C, приведенные в главе 4, свидетельствуют о том, что у всех адресов определенной сети совпадают значения некоторых октетов адресов. Например, сеть класса B 172.16.0.0 состоит из IP-адресов, которые начинаются с части 172.16: 172.16.0.0, 172.16.0.1, 172.16.0.2 и так далее до 172.16.255.255. Другой пример: сеть класса A 10.0.0.0 включает все адреса, начинающиеся с 10.

Подсеть IP (IP subnet) — это подмножество сетей класса A, B или C. На самом деле термин *подсеть* — это сокращение от *подразделенная сеть*. Например, одна подсеть сети класса B 172.16.0.0 могла бы быть набором всех IP-адресов, которые начинаются с 172.16.1 и включают адреса 172.16.1.0, 172.16.1.1, 172.16.1.2 и так далее до 172.16.1.255. Другая подсеть той же сети класса B могла бы включать все адреса, начинающиеся с 172.16.2.

Чтобы дать общее представление, на рис. 11.1 приведена базовая документация окончательного проекта подсети, полученная после разделения сети класса B 172.16.0.0.

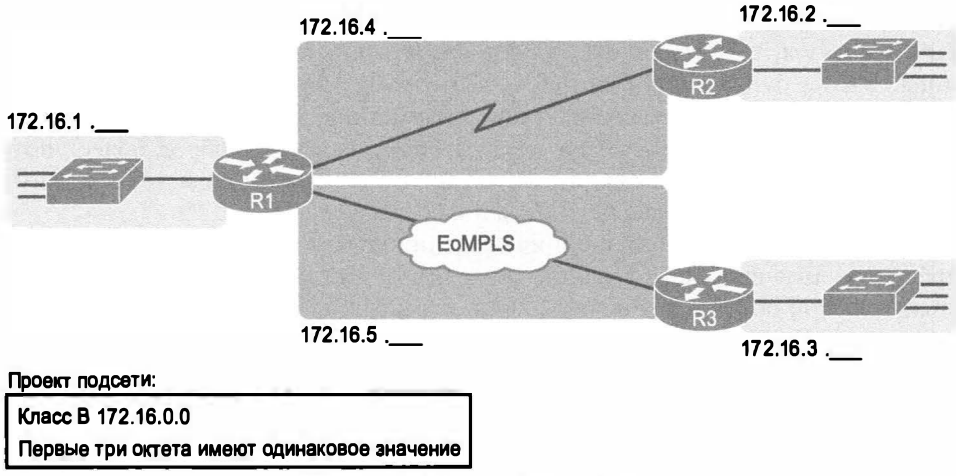


Рис. 11.1. Пример проекта подсети

Проект демонстрирует пять подсетей: по одной для каждой из трех локальных сетей и по одной для каждого из двух каналов WAN. Текстовые примечания — это пояснения, используемые инженером для подсетей: каждая подсеть включает адреса с одинаковым значением первых трех октетов. Например, число 172.16.1.__ для сети LAN, показанной слева, означает "все адреса начиная с 172.16.1". Обратите также внимание на то, что в данном проекте используются не все адреса сети класса B 172.16.0.0, таким образом, инженер оставил достаточно много места для последующего роста.

Оперативный или проектный подход к созданию подсетей

Большинство информационных задач требует, чтобы подсети создавались с учетом оперативного представления. Таким образом, прежде чем вы получите свою задачу, некто другой разработает то, как будет работать IP-адресация и подсети в данной конкретной корпоративной сети. Когда вы создаете подсеть, необходимо интерпретировать то, что некто уже выбрал.

Чтобы полностью понять IP-адресацию и подсети, необходимо рассмотреть их как с точки зрения проектного подхода, так и оперативного. Например, рис. 11.1 утверждает, что первые три октета во всех этих подсетях совпадают. Некий инженер уже выбрал проект, но почему именно этот? Какие альтернативы существуют? Может быть, эти альтернативы были бы сейчас лучше для данной объединенной сети? Все эти вопросы имеют отношение больше к проектному подходу создания подсетей, а не к оперативному.

Чтобы помочь оценить обе точки зрения, в некоторых главах этой части больше внимания уделяется проблемам проектирования, а не другим операциям при интерпретации некоего существующего проекта. В данной главе описан весь процесс проектирования, чтобы представить всю картину создания подсетей IP. В остальной

ных главах этой части каждая тема данной главы будет описана подробно либо с точки зрения оперативного подхода, либо проектного.

В трех остальных разделах текущей главы исследуется каждый из этапов, показанных на рис. 11.2.

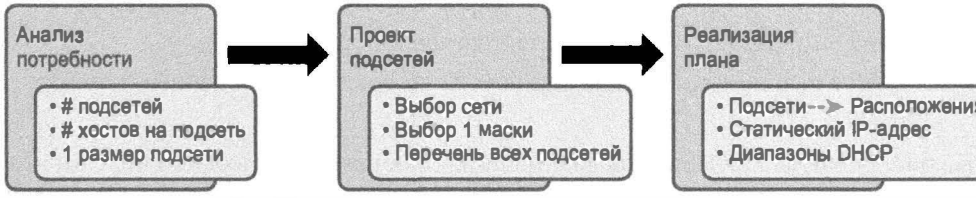


Рис. 11.2. Планирование, проектирование и реализация подсети

ВНИМАНИЕ!

В этой главе демонстрируется набор функций, вовлеченных в формальный процесс проектирования Cisco, называемый *подготовка, план, проект, реализация, работа и оптимизация* (Prepare, Plan, Design, Implement, Operate, Optimize — PPDIOO).

Анализ потребности в подсетях и адресации

В данном разделе обсуждается значение четырех простых вопросов, применяемых при анализе потребности в адресации и создании подсетей для любой новой или изменяющейся корпоративной сети.

1. Какие хосты должны группироваться в подсеть?
2. Сколько подсетей требует данная сеть?
3. Сколько IP-адресов хоста требует каждая подсеть?
4. Будет ли использован для простоты одинаковый размер подсети или нет?

Правила расположения хостов в определенной подсети

У каждого устройства, подключенного к объединенной сети IP, должен быть IP-адрес. К этим устройствам относятся компьютеры, используемые конечными пользователями, серверы, мобильные телефоны, портативные компьютеры, телефоны IP, планшеты и такие сетевые устройства, как маршрутизаторы, коммутаторы и брандмауэры. Короче говоря, в IP-адресе нуждается любое устройство, использующее протокол IP для передачи и получения пакетов.

ВНИМАНИЕ!

При обсуждении IP-адресации у термина *сеть* (network) есть вполне специфическое значение: сеть IP класса A, B или C. Во избежание недоразумений при использовании термина *сеть* при описании набора хостов, маршрутизаторов, коммутаторов и так далее в этой книге используются термины *объединенная сеть* (internetwork) и *корпоративная сеть* (enterprise network).

IP-адреса должны назначаться согласно неким простым правилам и на серьезных основаниях. Для эффективной работы маршрутизации и правил IP-адресации адреса группируют в группы, называемые подсетями. Эти правила приведены ниже.

Ключевая тема

Основные факты о подсетях

- Адреса в той же подсети не отделяются маршрутизатором.
- Адреса в различных подсетях разделены по крайней мере одним маршрутизатором.

На рис. 11.3 представлена общая концепция с хостами A и B в одной подсети и хостом C в другой. Обратите, в частности, внимание на то, что хосты A и B не отделены друг от друга никакими маршрутизаторами, а хост C отделен от хостов A и B по крайней мере одним маршрутизатором, следовательно, он находится в другой подсети.

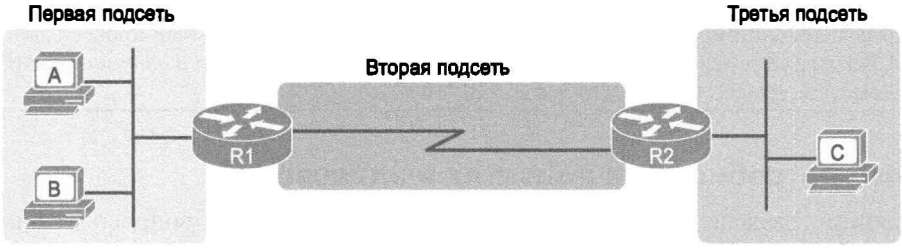


Рис. 11.3. Компьютеры A и B — в одной подсети, компьютер C — в другой

Концепция, согласно которой хост на том же канале должен находиться в той же подсети, очень похожа на концепцию почтового индекса. Все почтовые адреса в одном городе имеют одинаковый почтовый код (почтовый индекс). Адреса в другом городе, расположенном поблизости или на другом конце страны, имеют другой почтовый код. Почтовый код позволяет почтовой службе автоматизировать сортировку почты, чтобы доставлять ее в соответствующее место. По той же причине хосты одной локальной сети находятся в той же подсети, а хосты разных локальных сетей — в разных подсетях.

Обратите внимание на то, что двухточечный канал связи WAN на рисунке также нуждается в подсети. Маршрутизатор R1 на рис. 11.3 подключен к подсети LAN слева и к подсети WAN — справа. Маршрутизатор R2 подключен к той же подсети WAN. Для этого оба маршрутизатора, R1 и R2, будут иметь IP-адреса на своих интерфейсах WAN, и адреса эти будут в той же подсети. (Каналу связи WAN типа Ethernet поверх MPLS (EoMPLS) нужна та же IP-адресация на каждом из двух маршрутизаторов, имеющих IP-адрес в той же подсети.)

Локальные сети Ethernet на рис. 11.3 изображены немного иначе, с использованием простых линий без коммутаторов Ethernet. Когда подробности коммутаторов LAN не имеют значения, на схемах локальных сетей Ethernet все устройства изображают соединенными одной линией (см. рис. 11.3). (Это подражание кабельной проводке Ethernet, существовавшей до появления коммутаторов и концентраторов.)

И наконец, поскольку основная задача маршрутизаторов — перенаправлять пакеты из одной подсети в другую, маршрутизаторы обычно подключены к нескольким подсетям. В данном случае, например, маршрутизатор R1 соединен с одной

подсетью LAN слева и одной подсетью WAN — справа. Для этого маршрутизатор R1 будет настроен с двумя разными IP-адресами на каждом интерфейсе. Эти адреса будут находиться в разных подсетях, поскольку интерфейсы соединяют маршрутизатор с разными подсетями.

Определение количества подсетей

Чтобы определить количество требуемых подсетей, инженер должен обдумать документацию объединенной сети и применить приведенные ниже правила. Для этого ему нужен доступ к схемам сетей, подробностям конфигурации VLAN и, если используются глобальные сети WAN Frame Relay, подробности о *постоянных виртуальных каналах* (Permanent Virtual Circuit — PVC). На основании этой информации, используя соответствующие правила, можно спланировать одну или все подсети.

Ключевая тема

Какие места в сетевой топологии нуждаются в подсети

- Сеть VLAN.
- Двухточечный последовательный канал связи.
- Эмуляция Ethernet канала связи WAN (EoMPLS).
- Канал PVC Frame Relay

ВНИМАНИЕ!

Такие технологии WAN, как MPLS и Frame Relay, допускают создание подсетей и кроме одной подсети для канала WAN между двумя маршрутизаторами, но в этой книге рассматриваются технологии WAN, у которых есть только одна подсеть для каждого двухточечного соединения WAN между двумя маршрутизаторами.

Предположим, например, что для проектирования подсети у сетевого инженера есть только схема, приведенная на рис. 11.4.

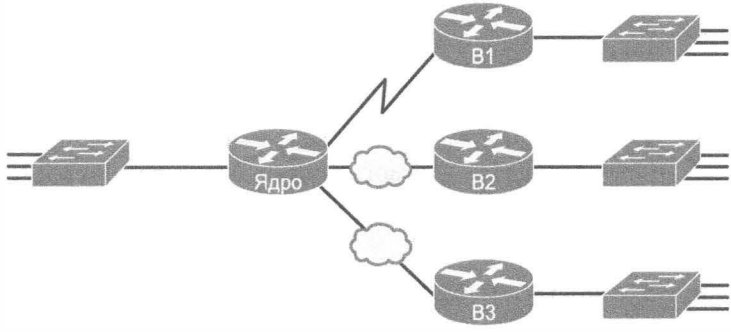


Рис. 11.4. Площадка из четырех объединенных сетей с маленькой центральной площадкой

На основании только этой схемы количество необходимых подсетей не может быть предсказано полностью. Конечно, три подсети будут необходимы для каналов связи сетей WAN, по одной на канал. Но каждый коммутатор сети LAN может быть подключен к одной сети VLAN или к нескольким. Как можно убедиться, на каждой площадке необходима по крайней мере одна подсеть для сети LAN, но может понадобиться и больше.

Теперь рассмотрим более подробную версию той же схемы, представленную на рис. 11.5. В данном случае на рисунке показано количество сетей VLAN в дополнение к равноправному уровню 3 топологии (маршрутизаторы и каналы связи, подключенные к маршрутизаторам). Здесь также показано, что у центральной площадки есть еще несколько коммутаторов, но ключевой факт представлен слева: независимо от количества имеющихся коммутаторов, центральная площадка имеет в общей сложности 12 сетей VLAN. Аналогично на рисунке каждая ветвь представлена как имеющая две сети VLAN. Наряду с теми же тремя подсетями WAN эта объединенная сеть требует 21 подсеть.

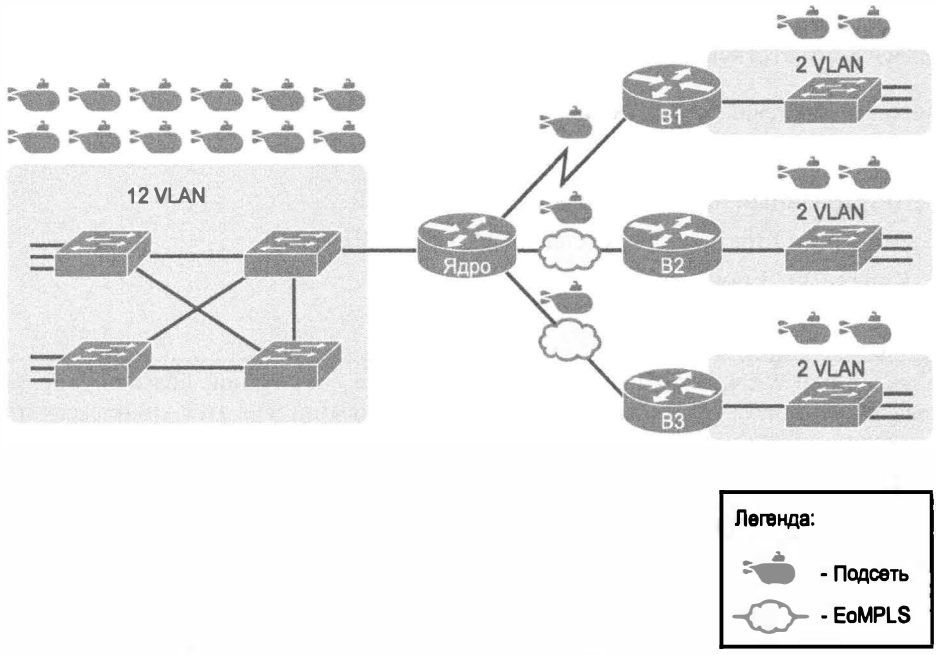


Рис. 11.5. Площадка из четырех объединенных сетей с большей центральной площадкой

И наконец, в реальном случае следовало бы рассмотреть нынешние потребности объединенной сети с учетом ее ожидаемого роста в будущем. Любой план подсетей должен включать реалистичную оценку количества подсетей, необходимых для удовлетворения будущих потребностей.

Определение количества хостов в каждой подсети

Определение количества хостов в подсети требует знания нескольких простых концепций, небольшого последующего исследования и опроса. Каждое устройство, подключенное к подсети, нуждается в IP-адресе. Для совершенно новой сети можно просмотреть бизнес-план — количество людей в подразделении, заявленные устройства — и таким образом получить некоторое представление о возможном количестве устройств. При расширении существующей сети, чтобы добавить новые площадки, можно использовать существующие площадки как объект для сравнения, а затем выяснить, какие площадки станут больше или меньше. И не забывайте

учитывать IP-адрес интерфейса маршрутизатора в каждой подсети и IP-адрес коммутатора для дистанционного управления им.

Вместо того чтобы собрать данные по каждой площадке, для планирования зачастую используют лишь несколько типичных площадок. Предположим, например, что имеется несколько больших коммерческих офисов и несколько меньших. В этом случае можно досконально изучить только один большой офис и только один малый. Добавьте в анализ тот факт, что магистральные линии в подсети нуждаются только в двух адресах, а также в большом количестве уникальных в своем роде подсетей, и вот достаточно информации, чтобы планировать проект адресации и подсетей.

Например, на рис. 11.6 показано, что инженер разработал схему, демонстрирующую количество хостов подсети LAN в наибольшей ветви, B1. Для двух других ветвей инженер не потрудился выяснить количество необходимых хостов. Пока количество необходимых IP-адресов площадок B2 и B3 остается ниже оценки 50, инженер может запланировать на основании большей площадки B1 по 50 хостов в каждой ветви подсети LAN и иметь достаточно много адресов на подсеть.

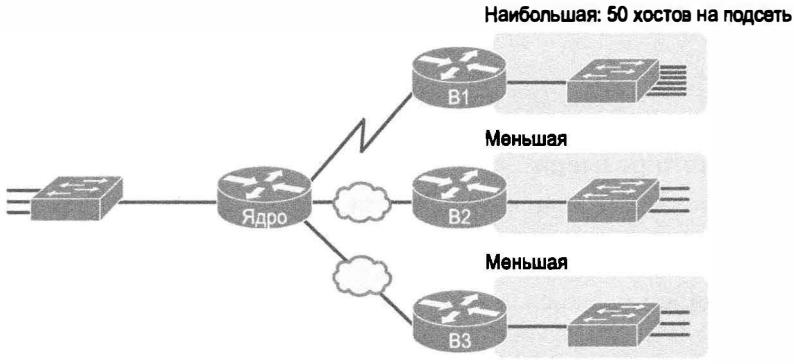


Рис. 11.6. Большая ветвь B1 с 50 хостами в подсети

Будут ли подсети одного размера

Последний выбор на начальном этапе планирования — будете ли вы использовать упрощенный проект по принципу "все подсети одного размера". Размер подсети (или ее длина) — это количество IP-адресов, пригодных для использования в подсети. Проект может подразумевать использование подсетей одинакового размера или разных размеров. У каждого из вариантов есть свои преимущества и недостатки.

Определение размера подсети

Прежде чем закончить чтение этой книги, вы изучите все подробности определения размера подсети, а пока вам достаточно знать лишь несколько специфических фактов о размере подсетей. Более подробная информация по этой теме приведена в главах 12 и 13.

Инженер назначает каждой подсети *маску подсети* (subnet mask), и эта маска, между прочим, как раз и определяет размер подсети. Маска резервирует количество *битов хоста* (host bit), задача которых — различать IP-адреса хостов в этой подсети. Поскольку с помощью x битов можно нумеровать 2^x сущностей, если маска определяет H битов хоста, подсеть может содержать 2^H индивидуальных числовых значений.

Однако размер подсети не 2^H, а $2^H - 2$, поскольку два числа в каждой подсети зарезервированы для других целей. Каждая подсеть резервирует наименьшее значение для *адреса подсети* (subnet number) и самое большое — для *широковещательного адреса подсети* (subnet broadcast address). В результате количество пригодных для использования IP-адресов в подсети составляет $2^H - 2$.

ВНИМАНИЕ!

Термины *номер подсети* (subnet number), *идентификатор подсети* (subnet ID) и *адрес подсети* (subnet address) описывают число, представляющее или идентифицирующее подсеть.

На рис. 11.7 представлена общая концепция трех частей структуры IP-адреса, с акцентом на часть хоста и результирующий размер подсети.

Ключевая тема

Рис. 11.7. Концепции размера подсети

Все подсети одного размера

Чтобы использовать в корпоративной сети подсети одного размера, достаточно применить одинаковую маску для всех подсетей, так как именно маска определяет размер подсети. Но какая маска?

При выборе единой маски следует учитывать одно требование: она должна обеспечить количество IP-адресов хостов, достаточное для поддержки наибольшей подсети. Для этого количество битов хоста (H), определенное маской, должно быть достаточно большим, чтобы значение $2^H - 2$ было большим или равным количеству IP-адресов хоста, необходимому в наибольшей подсети.

Рассмотрим, например, рис. 11.8. На нем показано необходимое количество хостов в подсети локальной сети. (На рисунке игнорируются подсети на каналах связи WAN, требующих только по два IP-адреса каждая.) Ветви локальной сети подсети требуют только по 50 адресов хоста, но основная подсеть локальной сети требует 200 адресов хоста. Для наибольшей подсети необходимо по крайней мере 8 битов хоста. Семи битов было бы недостаточно, поскольку $2^7 - 2 = 126$, а восьми битов хоста было бы вполне достаточно, поскольку $2^8 - 2 = 254$. Этого даже более чем достаточно для обеспечения 200 хостов в подсети.

В чем наибольшее преимущество при использовании подсетей одного размера? Оперативная простота. Другими словами, все остается простым. Сотрудники, обслуживающие сеть, легко привыкнут к работе с только одной одинаковой маской. Они легко ответят на все вопросы о создании подсетей, обсуждаемые в этой книге: вычисление идентификатора подсети и диапазона адресов в подсети, определение количества хостов в подсети и т.д. Это намного облегчает ответы на все вопросы по созданию подсетей, поскольку математические вычисления подсетей с одинаковой маской существенно проще.

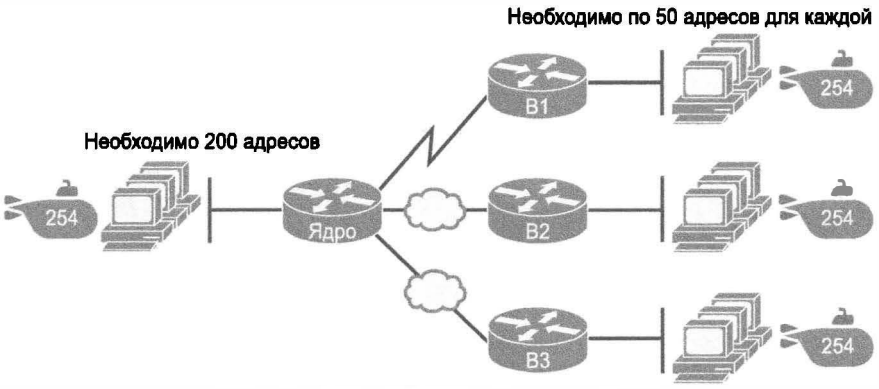


Рис. 11.8. Сеть с подсетями одного размера

Наибольшим недостатком единого размера подсетей является растрата впустую IP-адресов. Например, на рис. 11.8 показано, что все ветви подсети LAN обеспечивают 254 адреса, в то время как наибольшая подсеть нуждается только в 50 адресах. Подсети WAN нуждаются только в двух IP-адресах, но каждая поддерживает 254 адреса, снова растрачивая впустую большое количество IP-адресов.

Как бы то ни было, растрата впустую IP-адресов фактически не создает проблем в большинстве случаев. Большинство организаций в своих корпоративных объединенных сетях используют частные сети IP, и единая закрытая сеть класса A или B вполне может обеспечить множество IP-адресов, даже при потерях.

Подсети разного размера (маски подсети переменной длины)

Чтобы получить несколько подсетей разного размера в одной сети класса A, B или C, инженер должен создать одну подсеть, используя одну маску, другую — используя другую маску, и т.д. Различные маски означают различное количество битов хоста, используя результат формулы $2^H - 2$ для вычисления различного количества хостов в этих подсетях.

Рассмотрим, например, требования, перечисленные ранее на рис. 11.8. Там представлена одна подсеть LAN слева, требующая 200 адресов хостов, три ветви подсетей, которым требуется 50 адресов, и три последовательных канала, требующих по 2 адреса. Применение трех масок при создании трех подсетей разного размера, как показано на рис. 11.9, удовлетворит эти потребности с меньшей растратой впустую IP-адресов.

Меньшие подсети теперь растрачивают меньше IP-адресов по сравнению с прежним проектом на рис. 11.8. Подсети на рисунке справа, нуждающиеся в 50 IP-адресах, имеют подсети с 6 битами хоста, для $2^6 - 2 = 62$ доступных адресов на подсеть. Каналы WAN используют маски с 2 битами хоста, для $2^2 - 2 = 2$ доступных адресов на подсеть.

Но некоторые адреса все еще тратятся впустую, поскольку подсети нельзя задать некий произвольный размер. Размер всех подсетей вычисляется по формуле $2^H - 2$, где H — количество битов хоста, определенных маской для каждой подсети.

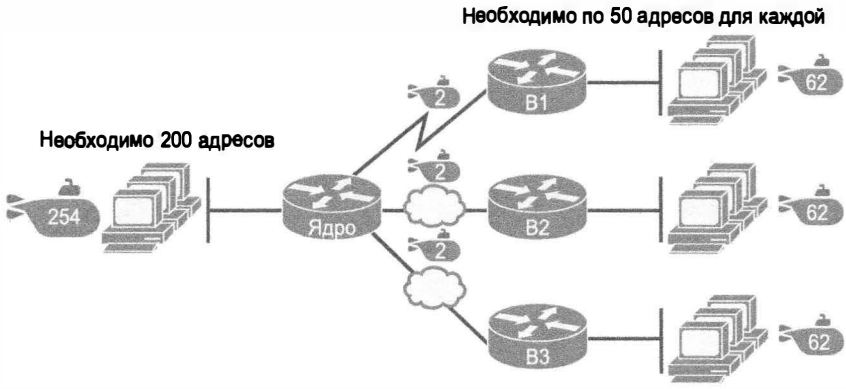


Рис. 11.9. Три маски, три размера подсети

В этой книге считается, что лучше иметь все подсети одинакового размера

Создание подсетей в этой книге рассматривается, по большей части, с использованием единой маски, чтобы все подсети имели одинаковый размер. Почему? Во-первых, это упрощает процесс обучения созданию подсетей. Во-вторых, некоторые типы анализа сети (а именно вычисление количества подсетей в классовой сети) возможны только при использовании единой маски.

Но все же следует быть готовым к работе с *масками подсети переменной длины* (Variable Length Subnet Mask — VLSM), подразумевающими практику использования различных масок для разных подсетей в той же классовой сети IP. Маскам VLSM посвящена глава 20, а в главе 21 рассматриваются также некоторые математические принципы, связанные с масками VLSM. Но во всех примерах вплоть до этих глав маски VLSM не используются только для простоты обучения.

Выбор проекта

Теперь, когда известно, как анализировать потребность в IP-адресации и подсетях, перейдем к следующему этапу — применению правил IP-адресации и созданию подсетей согласно этим потребностям и выбранным маскам. Другими словами, как создать конкретный проект подсетей, отвечающий этим требованиям, сколько необходимо подсетей и адресов хостов в наибольшей подсети? Короткий ответ — см. три задачи, представленные на рис. 11.10, *справа*.

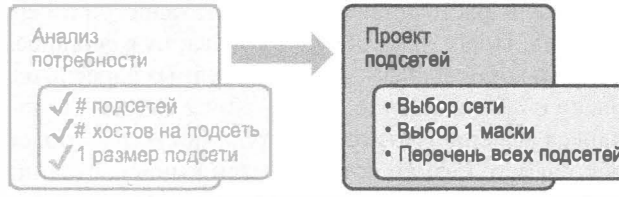


Рис. 11.10. Переход к этапу проектирования и от вопросов к ответам

Выбор классовой сети

В первоначальном проекте того, что ныне известно как Интернет, при реализации внутренней сети TCP/IP компании использовали зарегистрированные *открытые классовые сети IP* (public classful IP network). К середине 1990-х годов более популярными стали альтернативные *частные сети IP* (private IP network). В этом разделе обсуждаются причины выбора из этих двух альтернатив, поскольку он повлияет на выбор того, какую сеть IP компания будет впоследствии разделять на подсети и реализовывать в своей корпоративной объединенной сети.

Открытые сети IP

Первоначальный проект Интернета обязывал, чтобы любая подключившаяся к нему компания использовала *зарегистрированную открытую сеть IP* (registered public IP network). Для этого компания готовила некие документы, описывая объединенную сеть компании, количество существующих хостов, а также планы на ее рост. После передачи документов на рассмотрение компания получала сеть класса A, B или C.

Открытые сети IP и сопровождающие их административные процессы гарантируют, что все компании, которые подключаются к Интернету, будут использовать уникальные IP-адреса. В частности, как только компании присваивается открытая сеть IP, только эта компания должна использовать адреса в данной сети. Гарантия уникальности означает, что маршрутизация Интернета будет работать хорошо, поскольку нет никаких совпадающих открытых IP-адресов.

Рассмотрим, например, рис. 11.11. Компании 1 присвоена открытая сеть класса A 1.0.0.0, а компании 2 — открытая сеть класса A 2.0.0.0. Для открытой адресации в Интернете изначально предполагается, что после присвоения открытой сети никакие другие компании не смогут использовать адреса в сетях класса A 1.0.0.0 или 2.0.0.0.

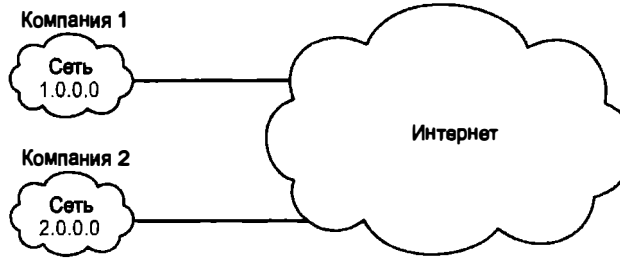


Рис. 11.11. Две компании с уникальными открытыми сетями IP

Этот первоначальный процесс присвоения адресов гарантировал уникальность IP-адресов по всей планете. Идея аналогична тому факту, что номер телефона должен быть уникальным в своей области, почтовый адрес и адрес электронной почты также должны быть уникальными. При вызове звонит телефон только того, кого вызывают, другие телефоны молчат. Аналогично компании 1 присвоена сеть класса A 1.0.0.0, и она назначает адрес 1.1.1.1 определенному компьютеру, причем этот адрес должен быть уникальным. Пакет, посланный через Интернет получателю 1.1.1.1, должен достичь только этого компьютера в компании 1 и не попасть на некий другой хост.

Исчерпание свободных IP-адресов

К началу 1990-х годов мир исчерпывал открытые сети IP, которые могли бы быть присвоены. Количество новых хостов, подключенных к Интернету, росло в темпе с двузначным числом *в месяц*. Компании продолжали следовать правилам, запрашивая открытые сети IP, и стало ясно, что текущая схема присвоения адресов не могла функционировать без небольшого изменения. Проще говоря, количества сетей класса A, B и C, обеспечиваемых 32-разрядным IP-адресом версии 4 (IPv4), оказалось недостаточно для предоставления одной открытой классовой сети на организацию, обеспечивая также достаточно много IP-адресов для каждой компании.

ВНИМАНИЕ!

Мир исчерпал открытые IPv4-адреса в начале 2011 года. Агентство IANA, которое присваивает блоки открытых IPv4-адресов пяти регистров Интернета в мире, присвоило последнее из пространств IPv4-адресов в начале 2011 года.

Сообщество Интернета упорно трудилось на протяжении 1990-х годов над решением этой проблемы и придумало несколько решений, включая перечисленные ниже.

Ключевая тема **Средства продления существования протокола IPv4**

- Новая версия протокола IP (IPv6) с намного бо́льшими адресами (128 битов).
- Назначение каждой компании части открытой сети IP вместо всей открытой сети IP.
- *Трансляция сетевых адресов* (Network Address Translation — NAT), позволяющая использовать частные сети IP.

Ныне эти три решения имеют большое значение для реальных сетей. Но чтобы не отклоняться от темы проектирования подсети, эта глава сосредоточивается на третьей возможности — частных сетях IP, которые применяются компаниями при использовании технологии NAT.

Технология NAT, подробно рассматриваемая в главе 24, позволяет нескольким компаниям использовать те же частные сети IP с теми же IP-адресами, что и у других компаний, при наличии подключения к Интернету. Например, на рис. 11.12 представлены те же две компании, подключенные к Интернету, что и на рис. 11.10, но теперь обе используют ту же частную сеть класса A 10.0.0.0.

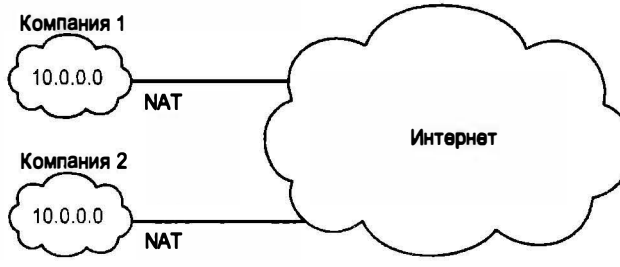


Рис. 11.12. Многократное использование той же закрытой сети 10.0.0.0 с помощью NAT

Обе компании используют ту же классовую сеть IP (10.0.0.0) и могут реализовать свой проект подсети внутренне, согласно собственным корпоративным объединенным сетям, не обсуждая их планы. Эти две компании могут даже использовать совпадающие IP-адреса в сети 10.0.0.0. В то же время, как ни удивительно, обе компании вполне могут даже общаться через Интернет.

Технология под названием *трансляция сетевых адресов* (Network Address Translation — NAT) позволяет компаниям многократно использовать те же сети IP, как показано на рис. 11.12. Технология NAT осуществляет это за счет трансляции IP-адресов в пакетах при их передаче от компании к Интернету, используя небольшое количество открытых IP-адресов для поддержки десятков тысяч частных IP-адресов. Этой сжатой информации недостаточно, чтобы понять, как работает NAT; но чтобы не отвлекаться от создания подсетей, отложим обсуждение работы технологии NAT до главы 23, а пока просто констатируем, что большинство компаний применяют технологию NAT и поэтому могут использовать частные сети IP для своих объединенных сетей.

Частные сети IP

Документ RFC 1918 определяет набор частных сетей IP, как указано в табл. 11.1. По определению эти частные сети IP:

- никогда не будут присваиваться организациям как открытая сеть IP;
- могут быть использованы организациями, применяющими технологию NAT при передаче пакетов в Интернет;
- могут использоваться организациями, которым никогда не придется посылать пакеты в Интернет.

Таким образом, при использовании технологии NAT (а почти все подключенные к Интернету организации используют NAT) компания вполне может выбрать одну или несколько частных сетей IP из зарезервированных номеров частных сетей. Документ RFC 1918 определяет список, который приведен в табл. 11.1.

Таблица 11.1. Документ RFC 1918. Частное пространство адресов

Класс сетей	Частные сети IP	Количество сетей
A	10.0.0.0	1
B	172.16.0.0 – 172.31.0.0	16
C	192.168.0.0 – 192.168.255.0	256

ВНИМАНИЕ!

Согласно неофициальному опросу, который я запустил на своем блоге в конце 2010 года, примерно половина посетителей указала, что у них используется частная сеть класса A 10.0.0.0, а не другая частная или открытая сеть.

Выбор сети IP на этапе проектирования

Ныне одни организации используют частные сети IP вместе с технологией NAT, а другие — открытые сети IP. Новые корпоративные объединенные сети используют частные IP-адреса с технологией NAT как часть соединения с Интернетом. Те орга-

низации, у которых уже есть зарегистрированные открытые сети IP, как правило, полученные до начала исчерпания адресов в 1990-х годах, продолжают использовать эти открытые адреса в своих корпоративных сетях.

Как только решено использовать частную сеть IP, остается лишь выбрать ту, у которой достаточно IP-адресов. У компании может быть маленькая объединенная сеть, но решено будет выбрать частную сеть класса A 10.0.0.0. Это может показаться расточительным, ведь у сети класса A более 16 миллионов IP-адресов, а нужно лишь несколько сотен. Но это не повлечет никакого штрафа и не создаст проблем с использованием закрытой сети, которая слишком велика для текущих или будущих потребностей.

В большинстве примеров этой книги используются адреса частных сетей IP. На этапе выбора количества сетей достаточно выбрать подходящую частную сеть класса A, B или C из списка закрытых сетей в документе 1918 RFC.

Независимо от математической и концептуальной точки зрения, методы разделения на подсети открытых и частных сетей IP одинаковы.

Выбор маски

Если бы разработчик следовал темам этой главы, то он знал бы следующее:

- необходимое количество подсетей;
- необходимое количество хостов на подсеть;
- что решено использовать только одну маску для всех подсетей, чтобы все подсети были одинакового размера (одинаковое количество хостов на подсеть);
- номер классовой сети IP, которая в результате будет разделена.

Этот раздел завершает описание процесса проектирования, по крайней мере, части, описанной в данной главе, обсуждением выбора одной маски, используемой для всех подсетей. Сначала для сравнения рассмотрим стандартные маски, используемые, когда сеть не разделена на подсети. Затем рассмотрим концепцию заимствования битов хоста для создания битов подсети. И наконец, закончим раздел примером создания маски подсети на основании анализа требований.

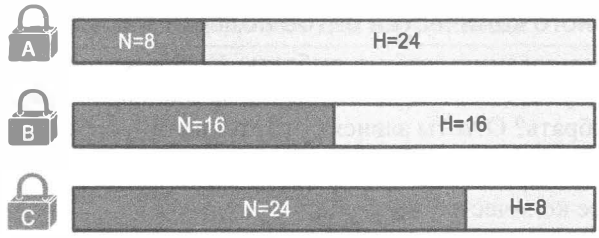
Классовые сети IP до создания подсетей

До разделения сети на подсети классовая сеть является единой группой адресов. Другими словами, инженер еще не разделил сеть на множество меньших подмножеств, называемых *подсетями*.

У адресов не разделенной классовой сети есть только две части: часть сети и часть хоста. Если сравнить два любых адреса классовой сети, то можно обнаружить следующее:

- у адресов одинаковое значение в части сети;
- у адресов различные значения в части хоста.

Настоящие величины частей сети и хоста адресов могут быть легко предсказаны, как показано на рис. 11.13.



Ключевая тема

Рис. 11.13. Формат адреса не разделенной на подсети сети класса A, B и C

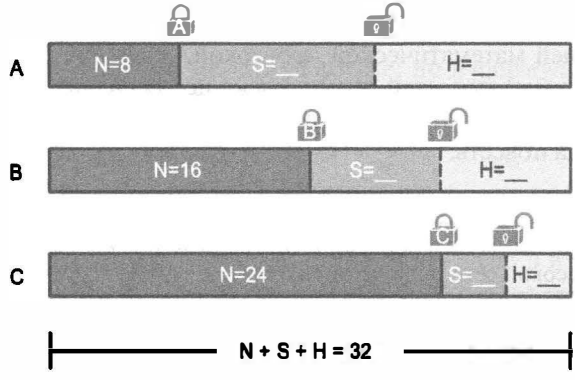
На рис. 11.13 значения N и H представляют количество битов сети и хоста соответственно. Правила классов определяют количество октетов сети (1, 2 или 3) для классов A, B и C соответственно; на рисунке эти значения представлены количеством битов. Количество октетов хоста составляет 3, 2 или 1 соответственно.

Продолжая анализ классовой сети перед созданием подсетей, можно вычислить количество классовых IP-адресов в сети по той же формуле, $2^{H} - 2$, что и ранее. В частности, размер не разделенных на подсети сетей класса A, B и C приведен ниже.

- **Класс A.** $2^{24} - 2 = 16\ 777\ 214$.
- **Класс B.** $2^{16} - 2 = 65\ 534$.
- **Класс C.** $2^{8} - 2 = 254$.

Заимствование битов хоста для создания битов подсети

Чтобы разделить сеть, инженеру следует обдумать части сети и хоста, как показано на рис. 11.13, а затем добавить посередине третью часть: часть подсети. Но он не может изменить размер сетевой части или размер всего адреса (32 бита). Чтобы создать часть подсети в структуре адреса, необходимые биты заимствуются из части хоста. Общая концепция представлена на рис. 11.14.



Ключевая тема

Рис. 11.14. Концепция заимствования битов хоста

Прямоугольники на рис. 11.14 означают части адреса. N — это количество битов сети. Остаются прямоугольники по 8, 16 битов или 24 бита, в зависимости от класса. Концептуально разработчик перемещает разделительную (пунктирную) линию в поле хоста между битами частей подсети (S) сетью и хоста (H), остающимися справа. В целом эти три части должны составить 32, поскольку IPv4-адреса состоят из 32 битов.

Выбор достаточного количества битов подсети и хоста

Процесс проектирования требует выбрать, где поместить пунктирную линию, представленную на рис. 11.14. Но какой выбор правильный? Сколько битов подсети и хоста следует выбрать? Ответы зависят от требований, собранных на прежних этапах процесса планирования:

- необходимое количество подсетей;
- количество хостов на подсеть.

Биты в части подсети позволяют уникально нумеровать различные подсети, которые разработчик хочет создать. С 1 битом подсети можно нумеровать 2^1 или 2 подсети. С 2 битами — 2^2 или 4 подсети, с 3 битами — 2^3 или 8 подсетей и т.д. Количество битов подсети должно быть достаточно большим для уникальной нумерации всех подсетей, как определено на этапе планирования.

В то же время количество оставшихся битов хоста должно быть достаточно большим для нумерации IP-адресов хостов в наибольшей подсети. Помните: в этой главе подразумевается использование одной маски для всех подсетей. Эта единая маска должна обеспечить и необходимое количество подсетей, и необходимое количество хостов в наибольшей подсети (рис. 11.15).

Ключевая тема

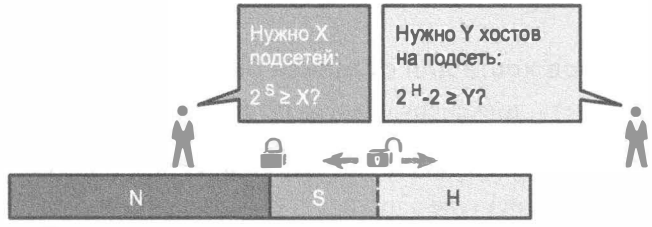


Рис. 11.15. Заимствование достаточного количества битов подсети и хоста

На рис. 11.15 представлена концепция выбора количества битов подсети (S) и хоста (H) с последующей математической проверкой. Значение 2^S должно быть больше количества необходимых подсетей, иначе маска не обеспечит достаточно много подсетей в этой сети IP. Кроме того, значение $2^H - 2$ должно быть больше количества необходимых хостов на подсеть.

ВНИМАНИЕ!

Идея вычисления количества подсетей как 2^S применима только в тех случаях, когда для всех подсетей единой классовой сети используется одна маска, как подразумевается в этой главе.

Для эффективной разработки или интерпретации маски, выбранной кем-то еще, необходимо хорошо помнить степени числа 2. В табл. 11.2 приведен список степеней числа 2 до 2^{12} наряду со столбцом $2^H - 2$, полезным при вычислении количества хостов на подсеть. В приложении Б приведена таблица со степенями числа 2 до 2^{24}.

Пример проекта: 172.16.0.0, 200 подсетей, 200 хостов

Для подкрепления теоретического обсуждения рассмотрим пример выбора маски подсети. В данном случае план и выбранный проект диктуют следующее:

Таблица 11.2. Степени числа 2. Справочник для выбора маски

Количество битов	2^x	$2^x - 2$
1	2	0
2	4	2
3	8	6
4	16	14
5	32	30
6	64	62
7	128	126
8	256	254
9	512	510
10	1024	1022
11	2048	2046
12	4096	4094

- использовать одну маску для всех подсетей;
- наличие 200 подсетей;
- наличие 200 IP-адресов хостов на подсеть;
- использовать частную сеть класса B 172.16.0.0.

Для выбора маски следует подумать о том, сколько битов подсети (S) необходимо для нумерации 200 подсетей?

Как можно заметить в табл. 11.2, значение S = 7 не является достаточно большим ($2^7 = 128$), но S = 8 вполне достаточно ($2^8 = 256$). Таким образом, для подсети необходимо по крайней мере 8 битов.

Затем, на основании количества хостов в подсети, следует задаться вопросом: сколько битов хоста (H) необходимо для нумерации 200 хостов в подсети?

В принципе математический механизм тот же, но в формуле расчета количества хостов на подсеть вычитается 2. Как можно заметить в табл. 11.2, значение H = 7 не является достаточно большим ($2^7 - 2 = 126$), но значения H = 8 вполне достаточно ($2^8 - 2 = 254$).

Всем требованиям в данном случае отвечает только одна возможная маска. Количество битов сети (N) должно быть 16, так как проект использует сеть класса B. Согласно требованиям, маска нуждается по крайней мере в 8 битах подсети, и по крайней мере в 8 битах хоста. Маски имеют только 32 бита; полученная маска представлена на рис. 11.16.

Маски и их форматы

Хотя инженеры считают, что IP-адреса состоят из трех частей (сеть, подсеть и хост), при выборе проекта маска подсети предоставляет инженеру способ распространить сделанный выбор на все устройства в подсети.

Маска подсети — это 32-разрядное двоичное число с несколькими двоичными единицами слева и двоичными нулями справа. По определению количество двоичных нулей равно количеству битов хоста. Фактически именно так маска выражает

идею размера части хоста в адресе при наличии подсети. Начальные биты в маске равны двоичным единицам, эти позиции двоичного разряда представляют совместно части сети и подсети в адресе при наличии подсети.

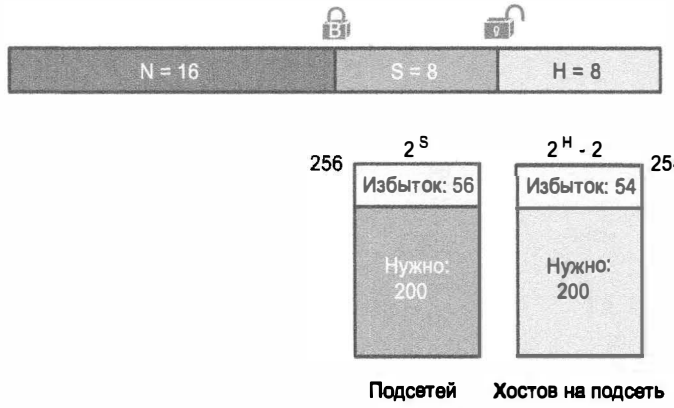


Рис. 11.16. Пример выбора маски N = 16, S = 8, H = 8

Поскольку сетевая часть всегда стоит на первом месте, затем идут часть подсети и часть хоста, маска подсети в бинарной форме не может чередовать единицы и нули. У каждой маски подсети есть одна неизменная строка двоичных единиц слева, а остальная часть битов занята двоичными нулями.

После выбора классовой сети и количества битов подсетей и хоста в подсети создать двоичную маску подсети несложно. Достаточно написать N единиц, S единиц, а затем H нулей (под N, S и H подразумевается количество битов сети, подсети и хоста). На рис. 11.17 представлена маска на основании условий предыдущего примера, в котором сеть класса B разделяется на подсети при 8 битах подсети и 8 битах хоста.

Рис. 11.17. Создание двоичной маски подсети для сети класса B

Кроме двоичной маски, представленной на рис. 11.17, маски могут быть также записаны в двух других форматах: уже знакомое *десятичное представление с разделительными точками* (Dotted-Decimal Notation — DDN), применяемое в IP-адресах, а также более краткая *префиксная* (prefix) форма. Эти форматы, а также преобразования различных форматов подробнее обсуждаются в главе 13.

Создание списка всех подсетей

Последняя задача этапа проектирования подсети — определение фактических подсетей, создаваемых на основании всех сделанных ранее выборов. На прежних этапах проектирования был определен класс A, B или C используемой сети, выбрана единая маска подсети, обеспечивающая достаточно много подсетей и достаточно много IP-адресов хостов в каждой подсети. Но каковы эти подсети? Как идентифицировать или описать подсеть? Данный раздел отвечает на эти вопросы.

Подсеть состоит из группы последовательных чисел, большинство из которых (номеров) применяется как IP-адреса хостов. Однако каждая подсеть резервирует первые и последние номера в группе, и эти два номера не могут использоваться как IP-адреса. В частности, каждая подсеть содержит следующее.

Элементы, совместно определяющие подсеть

Ключевая тема

- *Адрес подсети* (subnet address), называемый также *идентификатором подсети* (subnet ID) или *номером подсети* (subnet number), — число, идентифицирующее подсеть. Это наименьший номер в подсети. Он не может использоваться как IP-адрес хоста.
- *Широковещательный адрес подсети* (subnet broadcast или subnet broadcast address), называемый также *направленным широковещательным адресом* (directed broadcast address), — последний, самый большой номер в подсети. Он также не может использоваться как IP-адрес хоста.
- *IP-адреса* (IP address) — все номера между идентификатором подсети и широковещательным адресом подсети, применяемые как IP-адреса хостов.

Рассмотрим, например, приведенный ранее случай, в котором результаты проектирования были следующими:

- сеть 172.16.0.0 (класс B);
- маска 255.255.255.0 (для всех подсетей).

Прибегнув к математике, можно вычислить определенные факты о каждой подсети, существующей в этой сети класса B. В данном случае в табл. 11.4 приведены десять первых таких подсетей. Затем следует пропуск множества подсетей и приведены две последние подсети (в цифровой форме — наибольшие).

Таблица 11.3. Десять первых подсетей, а также несколько последних для сети 172.16.0.0 и маски 255.255.255.0

Адрес подсети	IP-адреса	Широковещательный адрес
172.16.0.0	172.16.0.1 – 172.16.0.254	172.16.0.255
172.16.1.0	172.16.1.1 – 172.16.1.254	172.16.1.255
172.16.2.0	172.16.2.1 – 172.16.2.254	172.16.2.255
172.16.3.0	172.16.3.1 – 172.16.3.254	172.16.3.255
172.16.4.0	172.16.4.1 – 172.16.4.254	172.16.4.255
172.16.5.0	172.16.5.1 – 172.16.5.254	172.16.5.255
172.16.6.0	172.16.6.1 – 172.16.6.254	172.16.6.255
172.16.7.0	172.16.7.1 – 172.16.7.254	172.16.7.255
172.16.8.0	172.16.8.1 – 172.16.8.254	172.16.8.255
172.16.9.0	172.16.9.1 – 172.16.9.254	172.16.9.255
	Много пропущено...	
172.16.254.0	172.16.254.1 – 172.16.254.254	172.16.254.255
172.16.255.0	172.16.255.1 – 172.16.255.254	172.16.255.255

Имея номер сети и маску, для вычисления идентификаторов подсетей и других подробностей о всех подсетях следует применить математику. В реальной жизни большинство инженеров используют калькуляторы подсети или инструменты планирования подсети. Для экзаменов CCENT и CCNA необходимо быть готовым самостоятельно вычислить эту информацию (в главе 19 описано вычисление всех подсетей данной сети).

Реализация плана

Следующий этап, планирование реализации, является последним этапом перед фактической настройкой устройств при создании подсети. Сначала инженер должен выбрать, где использовать каждую подсеть. Какую, например, из указанных в табл. 11.3 подсетей следует использоваться для каждой VLAN филиалов в некотором городе? Кроме того, какие из IP-адресов должны быть статическими, а какие можно использовать случайно? И наконец, какой диапазон IP-адресов в каждой подсети должен быть настроен на сервере DHCP и динамически предоставляться хостам для использования в качестве их IP-адресов? На рис. 11.18 приведен краткий список задач планирования реализации.

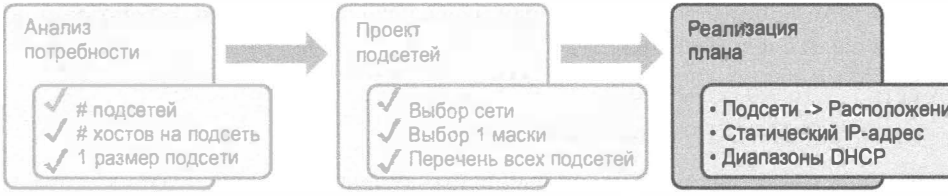


Рис. 11.18. Факты, сопутствующие этапу реализации плана

Назначение подсетей различным местам

Задача проста: просмотрите свою схему сети, выявите каждое расположение, которое нуждается в подсети, и выберите для них из таблицы по одной из возможных подсетей. Затем запишите, чтобы не забыть, в электронной таблице или другом специализированном инструменте планирования подсетей, какие подсети вы используете и где. Вот и все! На рис. 11.19 приведен пример законченного проекта, согласно табл. 11.3 и исходному проекту из примера, представленного на рис. 11.1.

Хотя в этом проекте вполне возможно использовать пять любых подсетей из табл. 11.3, в реальных сетях обычно придерживаются некой более осмысленной стратегии назначения подсетей. Например, можно было бы назначить всем подсетям LAN более низкие номера, а подсетям WAN более высокие. Либо можно было выделить большие диапазоны подсетей для различных подразделений. Либо можно было следовать той же стратегии, но игнорировать организационное деление в компании, уделяя больше внимания географии.

Например, для компании, расположенной в основном в Америке и с меньшим присутствием в Европе и Азии, можно было бы зарезервировать диапазоны подсетей на основании континентов. Такой выбор особенно полезен при последующей попытке использовать средство под названием *суммирование маршрутов* (route summarization), которое подробно обсуждается в главе 21. Рис. 11.20 дает общее представление об использовании подсетей, описанных в табл. 11.4.

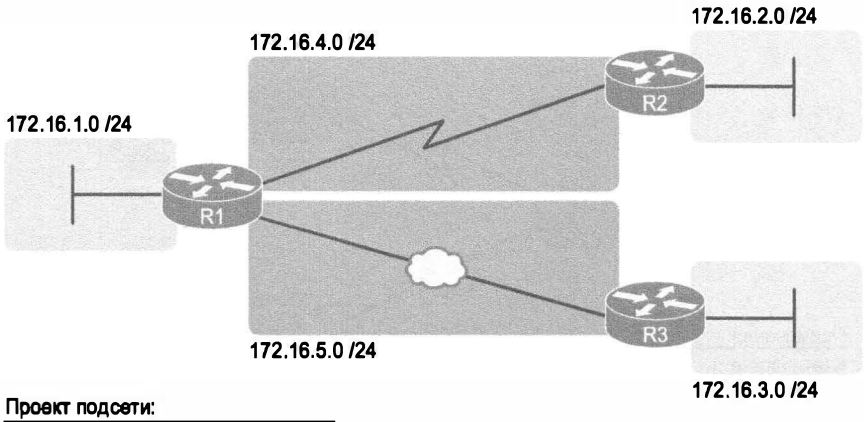


Рис. 11.19. Пример подсетей, назначенных различным расположениям

Рис. 11.20. Резервирование 50% подсетей для США и по 25% для Европы и Азии

Выбор статических и динамических диапазонов в подсети

Устройства получают свой IP-адрес и маску одним из двух способов: динамически, с помощью протокола DHCP, или статически, через конфигурацию. Чтобы протокол DHCP работал, сетевой инженер должен указать серверу DHCP подсети, для которых он должен назначить IP-адреса. Кроме того, эта конфигурация ограничивает сервер DHCP только неким подмножеством адресов в подсети. Для статических адресов можно просто настроить устройство, указав ему используемый IP-адрес и маску.

Чтобы по возможности не усложнять ситуацию, как правило, используется стратегия отделения статических IP-адресов с одной стороны диапазона адресов подсети и адресов DHCP, назначаемых динамически, с другой стороны. На самом деле не имеет значения, находятся ли статические адреса на нижнем конце диапазона адресов или на верхнем.

Предположим, например, что инженер решает выделить для подсетей LAN (см. рис. 11.19) пул DHCP, начинающийся сверху диапазона (а именно с адреса, заканчивающегося на .254) до .101. (Адрес, заканчивающийся на .255, естественно, зарезервирован.) Инженер выбирает статические адреса с нижнего конца диапазона (с адреса, заканчивающегося на .1) до .100. Идея представлена на рис. 11.21.

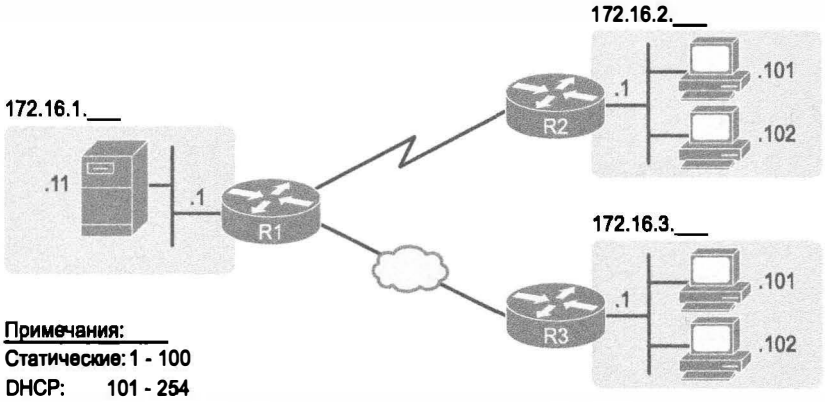


Рис. 11.21. Статические адреса — нижние, DHCP — верхние

Все три маршрутизатора на рис. 11.21 имеют статические IP-адреса, завершающиеся на .1. Единственный другой статический IP-адрес на рисунке присвоен серверу в левой части рисунка, это адрес 172.16.1.11 (сокращен на рисунке как .11).

В каждой сети LAN на рисунке справа есть по два компьютера, для динамического назначения IP-адресов которых используется сервер DHCP. Серверы DHCP, как правило, назначают адреса, начиная снизу диапазона адресов. Таким образом, хосты в каждой сети LAN получили бы адреса, заканчивающиеся на .101 и .102, что соответствует нижнему концу диапазона, выбранного в соответствии с проектом.

Обзор

Резюме

- Подсеть IP — это просто подмножество сетей класса A, B или C. На самом деле термин подсеть — это сокращение от подразделенная сеть.
- У каждого устройства, подключенного к объединенной сети IP, должен быть IP-адрес. К этим устройствам относятся компьютеры, используемые конечными пользователями, серверы, мобильные телефоны, портативные компьютеры, телефоны IP, планшеты и такие сетевые устройства, как маршрутизаторы, коммутаторы и брандмауэры. Короче говоря, в IP-адресе нуждается любое устройство, использующее протокол IP для передачи и получения пакетов.
- Для эффективной работы маршрутизации и правил IP-адресации адреса группируют в группы, называемые подсетями. Эти правила приведены ниже.
  - Адреса в той же подсети не отделяются маршрутизатором.
  - Адреса в различных подсетях разделены по крайней мере одним маршрутизатором.
- Размер подсети (или ее длина) — это количество IP-адресов, пригодных для использования в подсети. Проект может подразумевать использование подсетей одинакового размера или разных размеров.
- В частности, размер не разделенных на подсети сетей класса A, B и C приведен ниже.
  - Класс A. $224 - 2 = 16\,777\,214$.
  - Класс B. $216 - 2 = 65\,534$.
  - Класс C. $28 - 2 = 254$.
- Решение о количестве битов подсети и хоста зависит от требований, собранных на следующих этапах планирования:
  - необходимое количество подсетей;
  - количество хостов на подсеть.
- Устройства получают свой IP-адрес и маску одним из двух способов: динамически, с помощью протокола DHCP, или статически, через конфигурацию. Чтобы протокол DHCP работал, сетевой инженер должен указать серверу DHCP подсети, для которых он должен назначить IP-адреса. Кроме того, эта конфигурация ограничивает сервер DHCP только неким подмножеством адресов в подсети. Для статических адресов можно просто настроить устройство, указав ему используемый IP-адрес и маску.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Хост А — это компьютер, подключенный к коммутатору SW1 и присвоенный сети VLAN 1. Чему из приведенного ниже обычно назначается IP-адрес в той же подсети, что и хост А? (Выберите два ответа.).

   А) Интерфейс WAN локального маршрутизатора.

   Б) Интерфейс LAN локального маршрутизатора.

   В) Все остальные хосты, подключенные к тому же коммутатору.

   Г) Другие хосты, подключенные к тому же коммутатору, а также к сети VLAN 1.

2. Почему формула расчета количества хостов в подсети ($2^n - 2$) требует вычитания 2 адресов хостов?

   А) Чтобы зарезервировать два адреса для избыточных стандартных шлюзов (маршрутизаторов).

   Б) Чтобы зарезервировать два адреса, необходимых для работы протокола DHCP.

   В) Чтобы зарезервировать адреса для идентификатора подсети и стандартного шлюза (маршрутизатора).

   Г) Чтобы зарезервировать адреса для широковещательного адреса подсети и идентификатора подсети.

3. Сеть класса В должна быть разделена на подсети так, чтобы в результате она обеспечила 100 подсетей по 100 хостов на подсеть. Какие из следующих ответов перечисляют подходящую комбинацию количеств битов сети, подсети и хоста? (Выберите два ответа.).

   А) Сеть = 16, подсеть = 7, хост = 7.

   Б) Сеть = 16, подсеть = 8, хост = 8.

   В) Сеть = 16, подсеть = 9, хост = 7.

   Г) Сеть = 8, подсеть = 7, хост = 17.

4. Какая из приведенных ниже сетей IP является частной? (Выберите два ответа.)

   А) 172.31.0.0.

   Б) 172.32.0.0.

   в) 192.168.255.0.

   Г) 192.1.168.0.

   Д) 11.0.0.0.

5. Какая из приведенных ниже сетей IP является открытой? (Выберите три ответа.)

   А) 9.0.0.0.

   Б) 172.30.0.0.

   В) 192.168.255.0.

   Г) 192.1.168.0.

   Д) 1.0.0.0.

6. Какие части структуры IP-адресов должны уже существовать в сети класса В 172.16.0.0 прежде, чем она будет разделена сетевым инженером на подсети? (Выберите два ответа.)

   А) Сети.

   Б) Подсети.

В) Хоста.

Г) Широковещания.

7. Сетевой инженер уделяет время обдумыванию всей сети класса B 172.16.0.0 и ее разделению на подсети. Затем он решает, как разделить эту сеть класса B на подсети, создает план адресации и подсетей на бумаге, демонстрируя свой выбор. Если сравнить его представление об этой сети до создания подсетей и после, то каковы будут изменения структуры частей адресов в этой сети?

А) Часть подсети станет меньше.

Б) Часть хоста станет меньше.

В) Часть сети станет меньше.

Г) Часть хоста будет удалена.

Д) Часть сети будет удалена.

8. Какой из следующих терминов не используется для обозначения одного числа в каждой подсети, обычно однозначно определяющего подсеть? (Выберите два ответа.)

А) Идентификатор подсети (subnet ID).

Б) Номер подсети (subnet number).

В) Широковещательный адрес подсети (subnet broadcast).

Г) Имя подсети (subnet name).

Д) Адрес подсети (subnet address)

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 11.4.

Таблица 11.4. Ключевые темы главы 11

Элемент	Описание	Страница
Список	Основные факты о подсетях	356
Список	Какие места в сетевой топологии нуждаются в подсети	357
Рис. 11.7	Концепции размера подсети	360
Список	Средства продления существования протокола IPv4	364
Рис. 11.13	Формат адреса не разделенной на подсети сети класса A, B и C	364
Рис. 11.14	Концепция заимствования битов хоста	367
Рис. 11.15	Заимствование достаточного количества битов подсети и хоста	368
Список	Элементы, совместно определяющие подсеть	371

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

подсеть (subnet), сеть (network), классовая сеть (classful network), маски подсети переменной длины (Variable-Length Subnet Masks — VLSM), часть сети (network part), часть подсети (subnet part), часть хоста (host part), открытая сеть IP (public IP network), частная сеть IP (private IP network), маска подсети (subnet mask)

Ответы на контрольные вопросы:

1 Б и Г. **2** Г. **3** Б и В. **4** А и В. **5** А, Г и Д. **6** А и В. **7** Б. **8** В и Г.

ГЛАВА 12

Анализ классовых сетей IPv4

При работе с сетью выявление проблем зачастую начинается с выяснения IP-адреса и маски. Нужно уметь определить только по IP-адресу несколько фактов о сети класса A, B или C, в которой он располагается. Эти факты могут быть полезны при диагностике некоторых сетевых проблем.

В этой главе описан ряд ключевых фактов о классовых сетях IP и объяснено, как выявить эти факты. Затем будут приведены некоторые практические задачи. Прежде чем переходить к следующей главе, следует попрактиковаться, пока не удастся выявлять все эти факты быстро и уверенно на основании IP-адреса.

В этой главе рассматриваются следующие экзаменационные темы
Поиск и устранение неисправностей Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Концепции классовых сетей

Предположим, вы на собеседовании при приеме на первую работу в отрасли IT. В ходе собеседования вам предложен IPv4-адрес и маска: 10.4.5.99, 255.255.255.0. Что вы можете сказать о классовой сети (в данном случае сети класса A), в которой располагается этот IP-адрес?

Данный раздел — первый из двух основных разделов главы, в котором приведен обзор концепций *классовых сетей IP* (classful IP network), другими словами, сетей класса A, B и C. В частности, в текущей главе объясняется, как, начиная только с одного IP-адреса, выявить факты, приведенные ниже.

- Класс (A, B или C).
- Стандартная маска.
- Количество октетов (битов) сети.
- Количество октетов (битов) хоста.
- Количество адресов хостов в сети.
- Идентификатор сети.
- Широковещательный адрес сети.
- Первый и последний адреса, допустимые для использования в сети.

Классовая сеть IPv4 и связанные с ней факты

Протокол IP версии 4 (IPv4) определяет пять классов адресов. Три класса, A, B и C, используют одноадресатные IP-адреса. *Одноадресатные адреса* (unicast address) идентифицируют один хост или интерфейс, таким образом, он однозначно идентифицирует устройство. Адреса класса D служат многоадресатными адресами; так, пакет, посланный на один многоадресатный IPv4-адрес класса D, фактически будет доставлен нескольким хостам. И наконец, адреса класса E являются экспериментальными.

Класс может быть идентифицирован на основании значения первого октета адреса, как показано в табл. 12.2.

Ключевая тема **Таблица 12.1. Классы IPv4-адресов на основании значения первого октета**

Класс	Значения первого октета	Назначение
A	1–126	Одноадресатный (большие сети)
B	128–191	Одноадресатный (сети среднего размера)
C	192–223	Одноадресатный (маленькие сети)
D	224–239	Многоадресатный
E	240–255	Экспериментальный

Вопросы экзаменов CCENT и CCNA сосредоточены главным образом на одноадресатных классах (A, B и C), а не на классах D и E. После идентификации класса сети как A, B или C множество других связанных с ними фактов можно воспроизве-

сти по памяти. В табл. 12.3 приведена информация для справки и последующего изучения; каждая из этих концепций описана в данной главе.

Таблица 12.2. Основные факты о классах A, B и C

Ключевая тема

Частные сети IP	Класс A	Класс B	Класс C
Диапазон первого октета	1 – 126	128 – 191	192 – 223
Допустимые адреса сети	1.0.0.0 – 126.0.0.0	128.0.0.0 – 191.255.0.0	192.0.0.0 – 223.255.255.0
Всего сетей	$2^7 - 2 = 126$	$2^{14} = 16\,384$	$2^{21} = 2\,097\,152$
Хостов на сеть	$2^{24} - 2$	$2^{16} - 2$	$2^8 - 2$
Октеты (биты) в части сети	1 (8)	2 (16)	3 (24)
Октеты (биты) в части хоста	3 (24)	2 (16)	1 (8)
Стандартная маска	255.0.0.0	255.255.0.0	255.255.255.0

Реальные сети класса A, B и C

В табл. 12.2 приведен список диапазонов адресов сетей класса A, B и C. Однако некоторые ключевые моменты могут отсутствовать в справочной таблице. В данном разделе исследуются адреса сетей классов A, B и C, акцентируя внимание на важных темах, исключениях и необычных случаях.

В первую очередь, количество сетей в каждом классе значительно отличается. В классе A существуют лишь 126 сетей: 1.0.0.0, 2.0.0.0, 3.0.0.0 и так далее до 126.0.0.0. В классе B — 16 384 сети, а в классе C — более 2 миллионов.

Далее, размер сети каждого класса также значительно отличается. Каждая сеть класса A относительно велика (более 16 миллионов IP-адресов хостов на сеть), поскольку первоначально они были предназначены для использования большими компаниями и организациями. Сети класса B меньше (более 65 тысяч хостов на сеть). И наконец, у сетей класса C, предназначенных для малых организаций, есть по 254 адреса хоста в каждой сети (рис. 12.1).

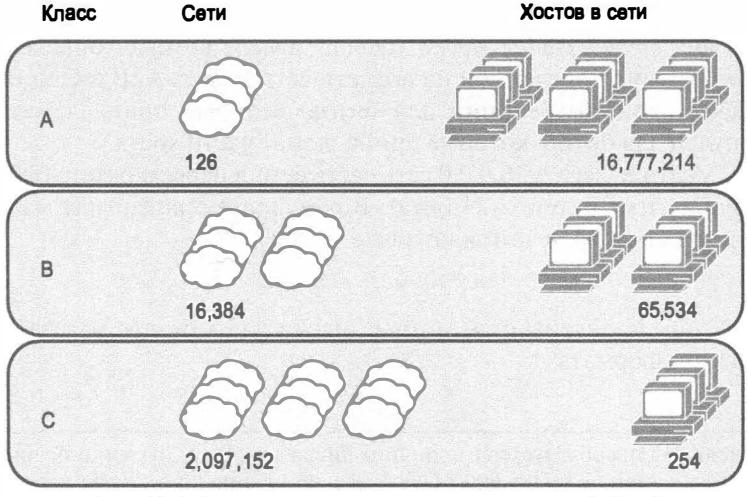


Рис. 12.1. Количество и размеры сетей класса A, B и C

Форматы адресов

Иногда сети класса A, B или C можно считать сетями, не разделенными на подсети. В таком случае у адресов классовой сети есть структура из двух частей: *часть сети* (network part), иногда называемая *префиксом* (prefix), и *часть хоста* (host part). Далее, сравнив два любых IP-адреса в одной сети, можно заметить следующее.

Ключевая тема **Сравнение частей сети и хоста адресов в той же классовой сети**

- У адресов в той же сети одинаковые значения в части сети.
- У адресов в той же сети разные значения в части хоста.

Например, в сети класса A 10.0.0.0, по определению, часть сети состоит из первого октета. В результате у всех адресов в части сети одинаковое значение, а именно 10 в первом октете. Если сравнить какие-нибудь два адреса в сети, у них будут разные значения в последних трех октетах (октетах хоста). Например, у IP-адресов 10.1.1.1 и 10.1.1.2 одинаковое значение (10) в части сети, но разные значения в части хоста.

На рис. 12.2 приведены формат и размеры (в битах) частей сети и хоста IP-адресов в сети класса A, B и C до того, как они будут разделены на подсети.

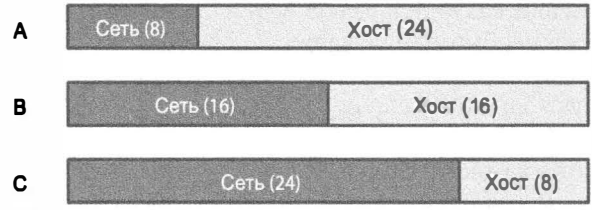


Рис. 12.2. Размеры (в битах) частей сети и хоста не разделенных на подсети классовых сетей

Стандартные маски

Хотя людям легко понять концепции, представленные на рис. 12.2, компьютеры предпочитают числа. Чтобы выразить эти же идеи для компьютера, с каждым классом сети связана *стандартная маска* (default mask), которая определяет размеры частей сети и хоста не разделенной на подсети сети класса A, B и C. Для этого маска содержит набор двоичных единиц для битов, которые принадлежат части сети, и двоичных нулей для битов, которые принадлежат части хоста.

Например, у сети класса A 10.0.0.0 есть часть сети в первом октете (8 битов) и часть хоста в последних трех октетах (24 бита). В результате стандартная маска (255.0.0.0) этого класса в двоичном виде выглядит так:

```
11111111 00000000 00000000 00000000
```

На рис. 12.3 представлены стандартные маски для каждого класса сети в двоичном и десятичном форматах.

ВНИМАНИЕ!

Десятичное число 255 преобразуется в двоичное число 11111111. Десятичное число 0 преобразуется в 8-битовое двоичное число 00000000. Вся таблица числовых преобразований приведена в приложении A.

Ключевая тема

Рис. 12.3. Стандартные маски для классов A, B и C

Количество хостов на сеть

Вычисление количества хостов в сети требует нескольких простых действий двоичной математики. Сначала рассмотрим случай, когда есть один двоичный знак. Сколько индивидуальных значений он обеспечит? Конечно, это два значения: 0 и 1. С двумя битами можно получить четыре комбинации: 00, 01, 10 и 11. Таким образом, имея N битов, можно получить 2^N уникальных комбинации.

Адреса хостов (IP-адреса, назначаемые хостам) должны быть уникальными. Биты хоста предназначены для предоставления каждому хосту уникального IP-адреса на основании разных значений в части хоста. Таким образом, имея H битов хоста, можно получить 2^H уникальных комбинации.

Однако количество хостов в сети не 2^H, а $2^H - 2$. Каждая сеть резервирует два числа, которые в противном случае могли бы использоваться как адреса хостов, но вместо этого используются для специальных целей: одно для идентификатора сети и одно для широковещательного адреса сети. Формула вычисления количества адресов хостов в сети класса A, B или C выглядит как $2^H - 2$, где H — количество битов хоста.

Ключевая тема

Определение идентификатора сети и сопутствующих значений

У каждой классовой сети есть четыре ключевых числа, которые описывают сеть. Эти четыре числа можно получить, исходя только из одного IP-адреса в сети. Это следующие значения:

- номер сети;
- первый (в цифровой форме самый низкий) пригодный для использования адрес;

- последний (в цифровой форме самый высокий) пригодный для использования адрес;
- широковещательный адрес сети.

Сначала рассмотрим номер сети и первый пригодный для использования IP-адрес. *Номер сети* (network number), называемый также *идентификатором сети* (network ID), или *адресом сети* (network address), идентифицирует сеть. По определению номер сети (в цифровой форме) — это самый малый номер. Однако для предотвращения любой двусмысленности разработчики IP-адресации добавили ограничение: номер сети не может быть назначен как IP-адрес. Таким образом, самый младший номер в сети — это ее идентификатор, а первый пригодный для использования в качестве IP-адреса — *на единицу больше, чем номер сети*.

Затем рассмотрим широковещательный адрес сети наряду с последним (в цифровой форме самым старшим) пригодным для использования IP-адресом. Документ RFC по протоколу TCP/IP определяет широковещательный адрес сети как специальный адрес в каждой сети. Этот широковещательный адрес применим как адрес получателя пакета, и маршрутизаторы отправляют копию пакета с таким адресом всем хостам в данной классовой сети. В цифровой форме широковещательный адрес сети — это всегда самый старший (последний) номер в сети. В результате самый старший (последний) номер, пригодный для использования в качестве IP-адреса, является адресом, который *на единицу меньше, чем широковещательный адрес сети*.

Проще говоря, если можно определить номер сети и широковещательный адрес сети, вычисление первого и последнего пригодных для использования IP-адресов в сети элементарно. На экзамене нужно уметь легко определить все четыре значения следующим образом.

Ключевая тема

Этапы определения информации о классовой сети

Этап 1	На основании первого октета определите класс сети (A, B или C)
Этап 2	На основании класса мысленно разделите октеты на части сети и хоста
Этап 3	Для выяснения номера сети замените октеты хоста IP-адреса на 0
Этап 4	Для выяснения первого адреса добавьте 1 к четвертому октету идентификатора сети
Этап 5	Для выяснения широковещательного адреса замените октеты хоста идентификатора сети на 255
Этап 6	Для выяснения последнего адреса вычтите 1 из четвертого октета широковещательного адреса сети

Описанный процесс выглядит трудней, чем он есть на самом деле. На рис. 12.4 приведен пример этого процесса для IP-адреса 10.1.2.3 сети класса A с соответствующими номерами этапов в кружочках.

На рис. 12.4 показана идентификация класса A сети (этап 1), а также количество октетов сети и хоста (1 и 3 соответственно). Чтобы выяснить идентификатор сети на этапе 3, скопирован только первый октет, последние три октета (хоста) заменены нулями. На этапе 4 скопирован только идентификатор сети и добавлена единица к четвертому октету. Аналогично для выяснения широковещательного адреса на этапе 5 скопированы октеты сети, а октеты хоста заменены на 255. Затем, на этапе 6, из

четвертого октета вычтена 1, чтобы выяснить последний (наибольший) пригодный для использования IP-адрес.

Рис. 12.4. Пример определения идентификатора сети и других значений для адреса 10.17.18.21

Чтобы продемонстрировать альтернативный пример, рассмотрим IP-адрес 172.16.8.9. На рис. 12.5 приведен процесс применительно к этому IP-адресу.

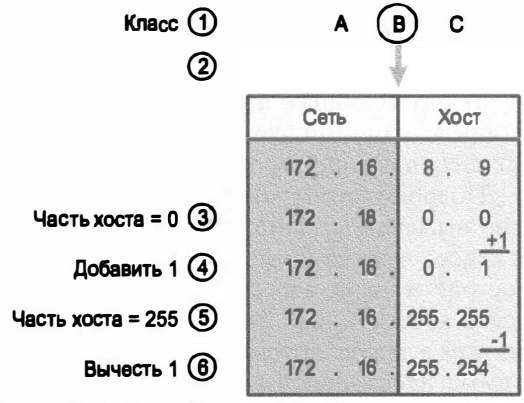


Рис. 12.5. Пример определения идентификатора сети и других значений для адреса 172.16.8.9

На рис. 12.5 показана идентификация сети класса B (этап 1), а также количество октетов сети и хоста (2 и 2 соответственно). Поскольку для выяснения идентификатора сети на этапе 3 скопированы только два первых октета, нулями заменены последние два октета (хоста). Аналогично этап 5 демонстрирует то же действие, но с установкой в 255 последних двух октетов (хоста).

Необычные идентификаторы сети и широковещательные адреса

Необычные номера из диапазона адресов сетей класса A, B и C, а также производные от них могут вызвать некоторое замешательство. В этом разделе приведены примеры некоторых необычно выглядящих, но вполне допустимых номеров.

Для класса A первый странный факт — это то, что в диапазоне значений первого октета отсутствуют числа 0 и 127. Кроме того, то, что могло бы быть сетью класса A 0.0.0.0, первоначально резервировалось для некоторых широковещательных целей, таким образом, все адреса, которые начинаются с 0 в первом октете, зарезервированы. То, что могло бы быть сетью класса A 127.0.0.0, все еще зарезервировано как специальный адрес, используемый при программной проверке, и называется *локальным диагностическим адресом* (loopback address) (127.0.0.1).

Для класса B (и C) некоторые из номеров сети могут выглядеть странно, особенно для тех, кто привык считать, что нули в конце означают идентификатор сети, а 255 в конце — широковещательный адрес сети. Во-первых, номера сети класса B располагаются в диапазоне от 128.0.0.0 до 191.255.0.0, в общей сложности 2^{14} сетей. Однако самый первый (младший) номер сети класса B (128.0.0.0) немного похож на номер сети класса A, поскольку он завершается тремя нулями. Но первый октет, 128, свидетельствует о том, что это сеть класса B с двумя октетами части сети (128.0).

Пример адреса 191.255.0.0 из верхней части диапазона сети класса B также может выглядеть странно на первый взгляд, но в действительности это самый верхний из допустимых адресов сети класса B. Широковещательный адрес такой сети (191.255.255.255) может немного походить на широковещательный адрес сети класса A из-за трех чисел 255 в конце, но в действительности это широковещательный адрес сети класса B.

Другие допустимые идентификаторы сети класса B, которые выглядят необычно, — это 130.0.0.0, 150.0.0.0, 155.255.0.0 и 190.0.0.0. Все они следуют соглашению о наличии значений от 128 до 191 в первом октете, значений от 0 до 255 во втором октете и двух или более нулей, поэтому они вполне допустимые идентификаторы сети класса B.

Сети класса C следуют тем же общим правилам, что и сети класса B, но с первыми тремя октетами, определяющими сеть. Номера сети находятся в диапазоне 192.0.0.0–223.255.255.0, с совпадающими значениями первых трех октетов для всех адресов в одной сети. Подобно сетям класса B, часть допустимых номеров сети класса C действительно выглядит странно. Например, сеть класса C 192.0.0.0 немного похожа на сеть класса A из-за последних трех октетов (0), но, поскольку это сеть класса C, она состоит из всех адресов, которые начинаются с трех октетов 192.0.0. Аналогично сеть класса C 223.255.255.0 — тоже вполне допустимая сеть класса C, состоящая из всех адресов, которые начинаются с 223.255.255.

Вот другие допустимые идентификаторы сети класса C, которые выглядят необычно: 200.0.0.0, 220.0.0.0, 205.255.255.0 и 199.255.255.0. Все они следуют соглашению о значении 192–223 в первом октете и значении 0–255 во втором и третьем октетах, а также нулем в четвертом октете.

Практические задания по классовым сетям

Подобно всем темам по IP-адресации и подсетям, на экзамене CCENT и CCNA необходимо быть готовым к практическим заданиям. Перед экзаменом следует овладеть концепциями и процессами, описанными в данной главе, а также быть в состоянии отвечать быстро и правильно. Не буду слишком подчеркивать важность владения IP-адресацией и подсетями для экзаменов, просто знайте эти темы, и знайте их хорошо.

Однако не стоит учить все в этой главе прямо сейчас. Необходимо немного попрактиковаться, чтобы удостовериться в понимании процессов. Можно пока использовать свои записи, эту книгу или все, что угодно. Достаточно попрактиковав-

шись, чтобы подтвердить способность получить правильные ответы, используя любую доступную помощь, темы этой главы станут понятны достаточно хорошо, чтобы можно было переходить к следующей главе.

Перед экзаменом попрактикуйтесь, чтобы овладеть темой данной главы полностью и отвечать достаточно быстро. Ключевые концепции и рекомендации такого двухэтапного подхода приведены в табл. 12.3.

Таблица 12.3. Продолжайте читать с учетом целей экзамена по темам данной главы

Период	Перед переходом к следующей главе	Перед сдачей экзамена
Сосредоточиться на ...	теме изучения	Быть быстрым и правильным
Разрешенные средства	Все	Ваш мозг и блокнот
Цель: точность	90% правильных ответов	100% правильных ответов
Цель: скорость	Любая скорость	10 секунд

Практические задания, следующие из ключевых фактов об IP-адресе

Примеры практических заданий по поиску различных фактов, следующие из ключевых фактов об IP-адресе, обсуждавшихся в этой главе, приведены в табл. 12.4. Попробуйте заполнить эту таблицу.

Таблица 12.4. Практическое задание: поиск идентификатора и широковещательного адреса сети

	IP-адрес	Класс	Количество октетов сети	Количество октетов хоста	Идентификатор сети	Широковещательный адрес сети
1	1.1.1.1					
2	128.1.6.5					
3	200.1.2.3					
4	192.192.1.1					
5	126.5.4.3					
6	200.1.9.8					
7	192.0.0.1					
8	191.255.1.47					
9	223.223.0.1					

Ответы приведены ниже, в разделе "Ответы на приведенные ранее практические задания".

Практическое задание на запоминание подробностей о классах адресов

Табл. 12.1 и 12.2, приведенные ранее в этой главе, содержали некую ключевую информацию о классах IPv4-адресов. Табл. 12.5 и 12.6 представляют собой не заполненные версии тех же таблиц. Попробуйте вспомнить эти ключевые факты, особенно диапазон значений в первом октете, который идентифицирует класс адресов, и заполнить эти таблицы. Затем вернитесь к табл. 12.1 и 12.2 и проверьте свои ответы. Повторяйте этот процесс до тех пор, пока не сможете запомнить всю информацию в таблицах.

Таблица 12.5. Не заполненная версия табл. 12.1

Значения первого октета	Класс	Назначение
	A	
	B	
	C	
	D	
	E	

Таблица 12.6. Не заполненная версия табл. 12.2

Частные сети IP	Класс A	Класс B	Класс C
Диапазон первого октета			
Допустимые адреса сети			
Всего сетей			
Хостов на сеть			
Октеты (биты) в части сети			
Октеты (биты) в части хоста			
Стандартная маска			

Дополнительные практические задания

Для дополнительной практики по классовым сетям можно использовать следующее.

- Приложение Г на веб-сайте, в котором содержатся дополнительные практические задачи. Оно содержит также объяснения по поиску ответа каждого задания.
- Создайте собственные задания. Можно случайно выбрать любой IP-адрес и попытаться выяснить ту же информацию, что и в практических заданиях этого раздела. Затем, чтобы проверить ответ, воспользуйтесь любым калькулятором подсетей. Большинство калькуляторов подсетей вычисляют класс и идентификатор сети.

Обзор

Резюме

- Протокол IP версии 4 (IPv4) определяет пять классов адресов. Три класса, A, B и C, используют одноадресатные IP-адреса, идентифицирующие один хост или интерфейс, таким образом, он однозначно идентифицирует устройство. Адреса класса D служат многоадресатными адресами; так, пакет, посланный на один многоадресатный IPv4-адрес класса D, фактически будет доставлен нескольким хостам. И наконец, адреса класса E являются экспериментальными.
- Структура адресов классовой сети состоит из двух частей: часть сети, иногда называемая префиксом, и часть хоста.
  - У адресов в той же сети одинаковые значения в части сети.
  - У адресов в той же сети разные значения в части хоста.
- С каждым классом сети связана стандартная маска, которая определяет размеры частей сети и хоста не разделенной на подсети сети класса A, B и C. Для этого маска содержит набор двоичных единиц для битов, которые принадлежат части сети, и двоичных нулей для битов, которые принадлежат части хоста.
- У каждой классовой сети есть четыре ключевых числа, которые описывают сеть. Эти четыре числа можно получить, исходя только из одного IP-адреса в сети. Это следующие значения:
  - номер сети;
  - первый (в цифровой форме самый низкий) пригодный для использования адрес;
  - последний (в цифровой форме самый высокий) пригодный для использования адрес;
  - широковещательный адрес сети.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Что из приведенного ниже не является допустимым идентификатором сети класса A? (Выберите два ответа.)

   A) 1.0.0.0.

   Б) 130.0.0.0.

   B) 127.0.0.0.

   Г) 9.0.0.0.

2. Что из приведенного ниже не является допустимым идентификатором сети класса B?

   A) 130.0.0.0.

   Б) 191.255.0.0.

В) 128.0.0.0.

Г) 150.255.0.0.

Д) Все это допустимые идентификаторы сети класса B.

3. Что из следующего является истиной об IP-адресе сети IP 172.16.99.45? (Выберите два ответа.)

А) Идентификатор сети 172.0.0.0.

Б) Сеть имеет класс B.

В) Для сети задана стандартная маска 255.255.255.0.

Г) В не разделенной на подсети сети для хостов предназначено 16 битов.

4. Что из следующего является истиной об IP-адресе сети IP 192.168.6.7? (Выберите два ответа.)

А) Идентификатор сети 192.168.6.0.

Б) Сеть имеет класс B.

В) Для сети задана стандартная маска 255.255.255.0.

Г) В не разделенной на подсети сети для хостов предназначено 16 битов.

5. Что из следующего является широковещательным адресом сети?

А) 10.1.255.255.

Б) 192.168.255.1.

В) 224.1.1.255.

Г) 172.30.255.255.

6. Что из следующего является идентификатором сети класса A, B или C?

А) 10.1.0.0.

Б) 192.168.1.0.

В) 127.0.0.0.

Г) 172.20.0.1.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 12.7.

Таблица 12.7. Ключевые темы главы 12

Элемент	Описание	Страница
Табл. 12.1	Классы IPv4-адресов на основании значения первого октета	380
Табл. 12.2	Основные факты о классах A, B и C	381
Список	Сравнение частей сети и хоста адресов в той же классовой сети	382
Рис. 12.3	Стандартные маски для классов A, B и C	383
Параграф	Формула вычисления количества адресов хостов в сети	383
Список	Этапы определения информации о классовой сети	384

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

сеть (network), классовая сеть (classful network), номер сети (network number), идентификатор сети (network ID), адрес сети (network address), широковещательный адрес сети (network broadcast address), первый адрес (first address), последний адрес (last address), часть сети (network part), часть хоста (host part), стандартная маска (default mask)

Практика

Если это еще не сделано, попрактикуйтесь в вопросах выявления деталей классовой сети, описанных в данной главе. Рекомендации приведены в разделе "Практические задания по классовым сетям".

Ответы на приведенные ранее практические задания

Набор практических заданий см. в табл. 12.3, а ответы приведены в табл. 12.8.

Таблица 12.8. Практическое задание: поиск идентификатора и широковещательного адреса сети

	IP-адрес	Класс	Количество октетов сети	Количество октетов хоста	Идентификатор сети	Широковещательный адрес сети
1	1.1.1.1	A	1	3	1.0.0.0	1.255.255.255
2	128.1.6.5	B	2	2	128.1.0.0	128.1.255.255
3	200.1.2.3	C	3	1	200.1.2.0	200.1.2.255
4	192.192.1.1	C	3	1	192.192.1.0	192.192.1.255
5	126.5.4.3	A	1	3	126.0.0.0	126.255.255.255
6	200.1.9.8	C	3	1	200.1.9.0	200.1.9.255
7	192.0.0.1	C	3	1	192.0.0.0	192.0.0.255
8	191.255.1.47	B	2	2	191.255.0.0	191.255.255.255
9	223.223.0.1	C	3	1	223.223.0.0	223.223.0.255

Чтобы определить класс, количество октетов сети и хоста, необходимо обратить внимание на первый октет IP-адреса. Если значение находится в диапазоне от 1 до 126 включительно, значит, адрес принадлежит сети класса A, с одним октетом сети и тремя октетами хоста. Если значение находится между 128 и 191 включительно, то адрес принадлежит сети класса B, с двумя октетами сети и двумя хоста. Если значение находится между 192 и 223 включительно — это адрес сети класса C, с тремя октетами сети и одним октетом хоста.

Значения последних двух столбцов находят на основании табл. 12.2, а именно количества октетов сети и хоста наряду с IP-адресом. Для поиска идентификатора сети скопируйте IP-адрес, но измените октеты хоста на 0. Аналогично для поиска широковещательного адрес сети скопируйте IP-адрес, но измените октеты хоста на 255.

Последние три задания могут вызвать сомнение, они были включены нарочно, чтобы продемонстрировать пример таких необычных случаев.

Ответ на практические задание 7 (см. табл. 12.3)

Рассмотрим IP-адрес 192.0.0.1. Первый октет, 192, находится ближе к нижней границе диапазона класса C; таким образом, у этого адреса есть три октета сети и один октет хоста. Для поиска идентификатора сети скопируйте адрес, но измените один октет хоста (четвертый) на 0. В результате получится 192.0.0.0. Выглядит странно, но это действительно идентификатор сети.

Нахождение широковещательного адреса сети для задания 7 также может выглядеть странно. Для этого скопируйте IP-адрес (192.0.0.1), но измените последний октет (единственный октет хоста) на 255. Получится широковещательный адрес 192.0.0.255. В частности, если покажется, что широковещательным адресом должен быть 192.255.255.255, значит, вы попали в ловушку. Сработала логика: "Замените все нули в идентификаторе сети на 255", что не является правильным. Вместо этого на 255 нужно изменить все октеты *хоста* в IP-адресе (или идентификаторе сети).

Ответ на практическое задание 8 (см. табл. 12.3)

Первый октет в задании 8 (191.255.1.47) находится в верхней части диапазона адресов сети класса B (128–191). Таким образом, для поиска идентификатора сети измените последние два октета (октеты хоста) на 0, получится 191.255.0.0. Это значение иногда создает проблемы, поскольку люди привыкли думать, что 255 означает широковещательный адрес.

Широковещательный адрес находится при замене двух октетов хоста на 255 и составляет 191.255.255.255. Выглядит скорее как широковещательный адрес сети класса A, но на самом деле это широковещательный адрес сети класса B 191.255.0.0.

Ответ на практическое задание 9 (см. табл. 12.3)

Последняя проблема с IP-адресом 223.223.0.1 заключается в том, что он ближе к верхней части диапазона адресов класса C. В результате, чтобы сформировать идентификатор сети 223.223.0.0, изменить на нуль нужно только последний октет (хоста). Выглядит похоже на номер сети класса B, поскольку заканчивается нулями в двух октетах, но это в действительности идентификатор сети класса C (на основании значения в первом октете)

Ответы на контрольные вопросы:

1 Б и В. **2** Д. **3** Б и Г. **4** А и В. **5** Г. **6** Б.

ГЛАВА 13

Анализ существующих масок подсети

Маска подсети, используемая в одной или нескольких подсетях объединенной сети IP, может много сказать о намерении разработчика подсети. В первую очередь, маска делит адреса на две части: *префикса* и *хоста*, где часть хоста определяет размер подсети. Кроме того, класс сети (A, B или C) далее делит структуру адресов на подсети, разделяя префиксную часть на части *подсети* и *сети*. Часть подсети определяет количество подсетей, способных существовать в одной классовой сети IP (подразумевается, что во всей классовой сети используется единая маска).

Маска подсети контролирует несколько важных моментов проекта подсетей. Однако, прежде чем приступать к анализу масок подсети, необходимо приобрести базовые математические навыки работы с масками. К ним также относится преобразование между тремя разными форматами представления масок:

- двоичное представление;
- десятичное представление с разделительными точками (Dotted-Decimal Notation — DDN);
- префиксное представление (называемое также CIDR).

Эта глава имеет два главных раздела. В первом основное внимание уделяется форматам масок и математическому механизму преобразования между ними. Второй раздел посвящен анализу значений IP-адреса и его маски подсети. В частности, она демонстрирует три части формата IPv4-адреса и описывает факты о проекте подсетей, определяемые маской.

В этой главе рассматриваются следующие экзаменационные темы

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Преобразование масок подсети

В этом разделе описано преобразование между различными форматами маски подсети. Впоследствии эти процессы можно использовать при решении практических задач. Если преобразования из одного формата в другой уже знакомы вам, переходите сразу к разделу "Практические задания по преобразованию масок подсети".

Три формата масок

Маски подсети могут быть записаны как 32-разрядные двоичные числа, но не любые. В частности, двоичная маска подсети должна следовать таким правилам.

Ключевая тема **Правила для двоичных значений маски подсети**

- Значение не должно чередовать единицы и нули.
- Если есть единицы, они располагаются слева.
- Если есть нули, они располагаются справа.

Например, следующие значения недопустимы. Первое недопустимо потому, что чередуются значения 0 и 1, а второе — потому, что 0 находятся слева, а 1 справа:

```
10101010 01010101 11110000 00001111
00000000 00000000 00000000 11111111
```

Следующие два двоичных значения отвечают требованиям, согласно которым все 1 находятся слева, а затем все 0 справа, без чередования 1 и 0:

```
11111111 00000000 00000000 00000000
11111111 11111111 11111111 00000000
```

Существуют еще два дополнительных формата маски подсети, чтобы не приходилось работать непосредственно с 32-битовыми двоичными числами. Один формат, *десятичное представление с разделительными точками* (Dotted-Decimal Notation — DDN), преобразует каждый набор из 8 битов в десятичный эквивалент. Например, две предыдущие двоичные маски можно преобразовать в следующие маски DDN, поскольку двоичные значения 11111111 преобразуются в десятичные 255, а двоичные 00000000 — в десятичное число 0.

```
255.0.0.0
255.255.255.0
```

Хотя формат DDN существовал с момента появления IPv4-адресации, третий, *префиксный* (prefix), формат маски был добавлен позже, в начале 1990-х годов. Этот формат использует правило, согласно которому маска подсети начинается с некоторого количества единиц, а остальные цифры являются нулями. Префиксный формат включает наклонную черту (/), сопровождаемую количеством двоичных единиц в двоичной маске. Используя те же два примера, что и ранее, получим следующие эквиваленты масок в префиксном формате:

```
/8
/24
```

Обратите внимание на то, что могут использоваться термины *префикс* (prefix), или *префиксная маска* (prefix mask), и *маска CIDR* (CIDR mask), или *маска наклонной черты* (slash mask). Этот более новый стиль префиксной маски появился одновременно со спецификацией *бесклассовой адресации* (Classless Interdomain Routing — CIDR) в начале 1990-х годов, и аббревиатуру CIDR ассоциировали со всем, что было связанно с адресацией CIDR, включая маски префиксного стиля. Кроме того, термин *маска наклонной черты* иногда используется потому, что значение метки включает наклонную черту (/).

И в реальной жизни, и на экзаменах Cisco CCENT и CCNA необходимо уметь работать с масками в любых форматах. В оставшейся части этого раздела рассматриваются преобразования между тремя форматами.

Преобразование между двоичным и префиксным форматами

Преобразование между двоичным и префиксным форматами маски должно быть относительно интуитивно понятным, поскольку известно, что префиксное значение — это просто количество двоичных единиц в двоичной маске. Для законченности изложения рассмотрим преобразование в каждом направлении.

Правила преобразования двоичных и префиксных форм маски

Ключевая тема

- **Из двоичной в префиксную**. Подсчитайте количество единиц в двоичной маске и запишите его в десятичной форме после /.
- **Из префиксной в двоичную**. Напишите количество единиц, соответствующее префиксному значению, и дополните их нулями до размера 32-разрядного двоичного числа.

В табл. 13.1 и 13.2 приведено несколько примеров.

Таблица 13.1. Пример преобразования: двоичная — в префиксную

Двоичная маска	Логика	Префиксная маска
11111111 11111111 11000000 00000000	Подсчитать 8 + 8 + 2 = 18 единиц	/18
11111111 11111111 11111111 11110000	Подсчитать 8 + 8 + 8 + 4 = 28 единиц	/28
11111111 11111000 00000000 00000000	Подсчитать 8 + 5 = 13 единиц	/13

Таблица 13.2. Пример преобразования: префиксная — в двоичную

Префиксная маска	Логика	Двоичная маска
/18	Написать 18 единиц, затем 14 нулей (всего 32)	11111111 11111111 11000000 00000000
/28	Написать 28 единиц, затем 4 нуля (всего 32)	11111111 11111111 11111111 11110000
/13	Написать 13 единиц, затем 19 нулей (всего 32)	11111111 11111000 00000000 00000000

Преобразование между двоичным форматом и DDN

По определению *десятичное число с разделительными точками* (DDN), используемое в IPv4-адресации, содержит четыре десятичных числа, отделенных точками. Каждое десятичное число представляет 8 битов. Так, одно число DDN представляет четыре десятичных числа, которые вместе представляют некое 32-разрядное двоичное число.

Преобразование маски из формата DDN в двоичный эквивалент относительно просто, но может оказаться трудоемко. Процесс преобразования описан ниже.

Для каждого октета осуществить преобразование из десятичной формы в двоичную.

Однако в зависимости от умения выполнять преобразование чисел из десятичной системы в двоичную этот процесс может быть трудоемким или длительным. Чтобы справиться с преобразованием масок на экзамене, выберите один из следующих методов преобразования и добейтесь его осуществления очень быстро и точно.

- Осуществляйте десятично-двоичные преобразования, но попрактикуйтесь, чтобы делать их быстро. Если решите выбрать этот путь, попробуйте игру Binary Game от Cisco, которую можно найти в учебной сети Cisco Learning Network (CLN) (`http://learningnetwork.cisco.com`).
- Используйте таблицу десятично-двоичных преобразований из приложения A. Это позволит находить ответы быстрее сейчас, но на экзамене ею воспользоваться не удастся.
- Запомните девять десятичных значений, которые могут быть в десятичной маске, и попрактикуйтесь в использовании справочной таблицы с этими значениями.

Третий метод является рекомендуемым методом в этой книге, он основан на том факте, что любой и каждый десятичный октет маски может иметь только одно из девяти значений. Почему? Помните, что двоичная маска не может чередовать 1 и 0, а 0 расположены справа? Этим правилам соответствуют только девять 8-битовых двоичных чисел. В табл. 13.3 приведен список этих значений, а также другая полезная информация.

Ключевая тема **Таблица 13.3. Девять значений, возможных в одном октете маски подсети**

Двоичный октет	Десятичный эквивалент	Количество двоичных единиц
00000000	0	0
10000000	128	1
11000000	192	2
11100000	224	3
11110000	240	4
11111000	248	5
11111100	252	6
11111110	254	7
11111111	255	8

Много процессов создания подсетей могут подразумевать использование двоичной математики, а могут и нет. Некоторые из них (включая преобразования маски) подразумевают использование информации из табл. 13.3. Необходимо запомнить информацию из этой таблицы. Рекомендуется сделать копию таблицы для удобства на время тренировки. (Вы, вероятно, запомните содержимое этой таблицы в процессе практики преобразований, причем вполне достаточно, чтобы осуществлять их быстро и правильно.)

С использованием таблицы процессы преобразования в каждом направлении с двоичными и десятичными масками осуществляются так.

Правила преобразования между двоичной и DDN формами маски

Ключевая тема

- **Из двоичной в десятичную.** Для каждого октета найдите в таблице двоичное значение и запишите соответствующее десятичное значение.
- **Из десятичной в двоичную.** Организуйте биты в четыре набора по восемь. Для каждого октета найдите в таблице десятичное значение и запишите соответствующее 8-битовое двоичное значение.

В табл. 13.4 и 13.5 приведено несколько примеров.

Таблица 13.4. Пример преобразования: двоичная — в десятичную

Двоичная маска	Логика	Десятичная маска
11111111 11111111 11000000 00000000	11111111 в 255 11000000 в 192 00000000 в 0	255.255.192.0
11111111 11111111 11111111 11110000	11111111 в 255 11110000 в 240	255.255.255.240
11111111 11111000 00000000 00000000	11111111 в 255 11111000 в 248 00000000 в 0	255.248.0.0

Таблица 13.5. Пример преобразования: десятичная — в двоичную

Десятичная маска	Логика	Двоичная маска
255.255.192.0	255 в 11111111 192 в 11000000 0 в 00000000	11111111 11111111 11000000 00000000
255.255.255.240	255 в 11111111 240 в 11110000	11111111 11111111 11111111 11110000
255.248.0.0	255 в 11111111 248 в 11111000 0 в 00000000	11111111 11111000 00000000 00000000

Преобразование между префиксным форматом и DDN

При обучении наилучший способ преобразования между префиксным и десятичным форматами подразумевает предварительное преобразование в двоичный формат. Например, чтобы перейти от десятичного числа к префиксному, сначала преобразуйте его в двоичное, а затем двоичное в префиксное.

При подготовке к экзамену добейтесь способности вычислять эти преобразования в уме. При обучении вы, вероятно, захотите использовать бумагу. Для тренировки попробуйте записывать не все в каждом октете двоичного числа, а только количество двоичных единиц.

На рис. 13.1 приведен пример преобразования префикса в десятичное число. Слева показан промежуточный этап преобразования в двоичный формат. Для срав-

нения справа приведен промежуточный этап преобразования в двоичный формат сокращено, где следует указать только количество двоичных единиц в каждом октете двоичной маски.

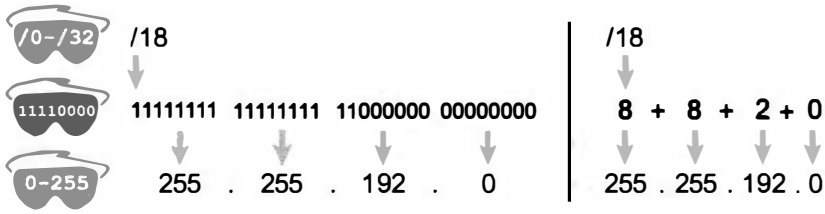


Рис. 13.1. Преобразование из префиксного формата в десятичный: полная и сокращенная формы

Аналогично при преобразовании десятичного числа в префикс мысленно преобразуйте его в двоичное, а по мере приобретения навыка используйте только количество единиц в каждом октете двоичного числа. На рис. 13.2 приведен пример такого преобразования.

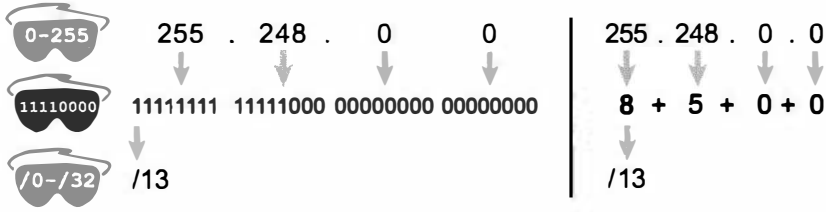


Рис. 13.2. Преобразование из десятичного формата в префиксный: полная и сокращенная формы

Обратите внимание: в приложении А есть таблица, в которой перечислены все 33 допустимые маски подсети во всех трех форматах.

Практические задания по преобразованию масок подсети

Прежде чем переходить ко второй части главы, рассмотрим, что означают маски подсети, и пройдем небольшую практику. Практикуйтесь в процессах, обсуждаемых в этой главе, пока не станете получать правильные ответы почти всегда. Используйте любые средства по своему усмотрению и любое время, чтобы прийти к соответствию с табл. 13.6 и быть готовым перейти к следующему разделу. Однако перед сдачей экзамена потренируйтесь, чтобы овладеть темой данной главы полностью и отвечать достаточно быстро. Ключевые концепции и рекомендации такого двухэтапного подхода приведены в табл. 13.6.

Список из восьми практических заданий приведен в табл. 13.7. У таблицы три столбца, по одному для каждого формата маски. В каждой строке содержится одна маска в одном из форматов. Ваша задача — найти значение маски в двух других форматах. Ответы приведены в табл. 13.10.

Таблица 13.6. Продолжайте читать с учетом целей экзамена по темам данной главы

Период	Перед переходом к следующей главе	Перед сдачей экзамена
Сосредоточиться на ...	теме изучения	Быть быстрым и правильным
Разрешенные средства	Все	Ваш мозг и блокнот
Цель: точность	90% правильных ответов	100% правильных ответов
Цель: скорость	Любая скорость	10 секунд

Таблица 13.7. Практическое задание: поиск значения маски в двух других форматах

Префиксная маска	Двоичная	Десятичная
	11111111 11111111 11000000 00000000	
		255.255.255.252
/25		
/16		
		255.0.0.0
	11111111 11111111 11111100 00000000	
		255.254.0.0
/27		

Дополнительные практические задания

Для дополнительной практики по преобразованию масок подсети можно использовать следующее.

- Приложение Д, в котором содержатся дополнительные практические задания. Оно содержит также объяснения по поиску ответа каждого задания.
- Создайте собственные задания. Поскольку существует только 33 корректных маски подсети, выберите любую и преобразуйте ее в два других формата. Затем проверьте ответ в приложении А, где перечислены все значения масок во всех трех форматах. (Рекомендация: преобразуйте префикс в двоичный формат, а затем в десятичный. Потом преобразуйте маску DDN в двоичный и префиксный форматы.)

Обратите внимание: эти преобразования требуются при решении многих разных задач на создание подсетей, поэтому дополнительная практика по этой теме гарантирована.

Выборов проекта подсети с использованием маски

У масок подсети много задач. Фактически, если десять опытных сетевых инженеров по отдельности спросить, "Какова цель маски подсети?", то будет получено множество истинных ответов. Маска подсети играет несколько ролей.

Эта глава посвящена одному специфическому способу использования маски подсети: определению префиксной части IP-адресов в подсети. Префиксная часть имеет одинаковое значение у всех адресов в подсети. Фактически подсеть может быть определена как набор из всех IPv4-адресов с одинаковым значением в префиксной части.

Хотя предыдущий абзац мог бы казаться немного формальным, сама идея относительно проста, как показано на рис. 13.3. Это схема сети с двумя подсетями: подсеть из всех адресов, начинающихся с 172.16.2, и вторая подсеть, состоящая из всех адресов, начинающихся с 172.16.3. В данном примере префиксе (часть с одинаковым значением во всех адресах подсети) состоит из первых трех октетов.

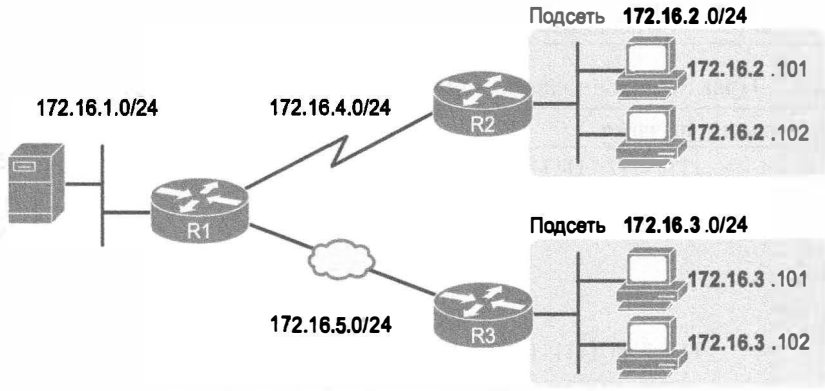


Рис. 13.3. Пример проекта подсети с маской /24

Люди вполне могут сесть за стол и решить, что префикс будет в три октета длиной, компьютеры для этого используют маску подсети. В данном случае, подсети используют маску /24, означающую, что префиксная часть адресов составляет 24 бита (3 октета).

Чтобы понять концепцию использования маски подсети префиксной части IPv4-адреса, наряду с этим другим использованием для маски подсети. Обратите внимание, что в этом разделе обсуждаются первые пять элементов в списке.

Этот раздел рассматривает не только использование маски подсети и концепцию префиксной части IPv4-адреса, но и другие ее назначения. Обратите внимание, что этот раздел посвящен первым пяти элементам в списке.

Фактически маска подсети используется для следующих целей.

Ключевая тема **Некоторые функции маски подсети**

- Определяет размер префиксной части (сети и подсети) адресов подсети.
- Определяет размер части хоста адресов подсети.
- Применяется при вычислении количества хостов в подсети.
- Позволяет сетевому инженеру выяснить подробности о проекте подсети (количество битов подсети и хоста).
- Согласно определению, используется при вычислении количества подсетей во всей классовой сети.
- Применяется в двоичных вычислениях идентификатора и широковещательного адреса подсети.

Маски делят адреса подсети на две части

Маска подсети разделяет IP-адреса подсети на две части: *префикса* (или *подсети*) и *хоста*.

Часть префикса идентифицирует адреса, которые располагаются в той же подсети, поскольку у всех IP-адресов в той же подсети одинаковое значение в префиксной части их адресов. Идея очень похожа на почтовый индекс в адресах обычной почты. У всех почтовых адресов в одном городе одинаковый почтовый индекс. Аналогично у всех IP-адресов в одной подсети идентичные значения в префиксной части адресов.

Часть хоста в адресе уникально идентифицирует хост в подсети. Если сравнить какие-нибудь два IP-адреса в той же подсети, то их части хоста будут отличаться, даже при том, что в префиксных частях их адресов то же значение. Подведем итог сравнения.

Сравнение IP-адресов в одной подсети

Ключевая тема

- **Часть префикса (подсети).** Одинаковы во всех адресах той же подсети.
- **Часть хоста.** Различны во всех адресах той же подсети.

Предположим, например, что есть подсеть, концептуально включающая все адреса с первыми тремя октетами 10.1.1. Несколько адресов этой подсети приведено ниже.

10.1.1.**1**
10.1.1.**2**
10.1.1.**3**

В этом списке части префикса или подсети (первые три октета 10.1.1) одинаковы, а части хоста (последний октет, выделенный полужирным шрифтом) разные. Таким образом, часть префикса или подсети адреса идентифицирует группу, а часть хоста — определенный элемент группы.

Маска подсети определяет разделительную линию между частями префикса и хоста. Для этого маска создает концептуальную линию между двоичными единицами и нулями. Короче говоря, если маска имеет P двоичных единиц, то префиксная часть имеет длину в P бит, а остальная часть битов является битами хоста. Обшая концепция представлена на рис. 13.4.

Ключевая тема

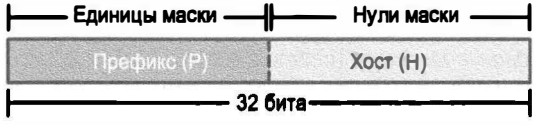


Рис. 13.4. Части префикса (подсети) и хоста, разделенные единицами и нулями маски

На рис. 13.5 представлен конкретный пример — использование маски 255.255.255.0. Маска 255.255.255.0 (/24) имеет 24 двоичных единицы, т.е. длина префикса составляет 24 бита.

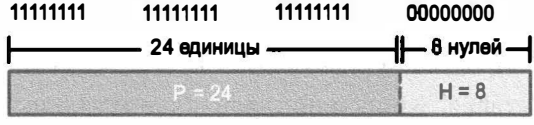


Рис. 13.5. Маска 255.255.255.0: P=24, H=8

Маски и класс делят адреса на три части

Кроме представления с двумя частями IPv4-адресов, возможно наличие трех частей. Для этого достаточно применить к формату адреса правила класса A, B и C, чтобы выявить часть сети в начале адреса. Эта дополнительная логика делит префикс на две части: часть *сети* и часть *подсети*. Класс определяет длину части сети, а часть подсети является остальной частью префикса. Концепция представлена на рис. 13.6.

Ключевая тема

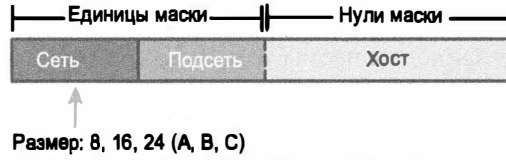


Рис. 13.6. Применение концепции класса для создания трех частей адреса

Совместно части сети и подсети составляют префикс, поскольку у всех адресов в той же подсети должны быть идентичные значения в частях подсети и сети. При рассмотрении адреса с точки зрения наличия двух или трех частей часть хоста остается неизменной.

Для полноты картины на рис. 13.7 демонстрируется тот же пример, что и в предыдущем разделе, с подсетью "все адреса, которые начинаются с 10.1.1". В этом примере подсеть использует маску 255.255.255.0, и все адреса принадлежат сети класса A 10.0.0.0. Класс определяет 8 битов сети, а маска — 24 бита префикса, значит, в подсети существует 24 – 8 = 16 битов. Часть хоста остается равной 8 битам, согласно маске.

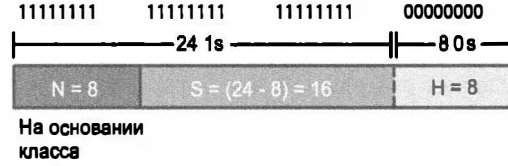


Рис. 13.7. Подсеть10.1.1.0, маска255.255.255.0: N=8, S=16, H=8

Бесклассовая и классовая адресация

Термины *бесклассовая адресация* (classless addressing) и *классовая адресация* (classful addressing) относятся к двум разным способам восприятия IPv4-адресов, как было описано в этой главе. Классовая адресация предполагает, что применяются правила классов A, B и C, таким образом, префикс разделяется на части сети и подсети, как на рис. 13.6 и 13.7. Бесклассовая адресация означает, что правила классов A, B и C игнорируются, т.е. префиксная часть считается единой частью, как показано на рис. 13.4 и 13.5. Следующие более формальные определения приведены для справки и изучения.

Ключевая тема

Определения классовой и бесклассовой адресации

- **Бесклассовая адресация.** Концепция наличия у IPv4-адреса двух частей (префиксной и хоста), определенных только маской, *без учета класса* (A, B или C).
- **Классовая адресация.** Концепция наличия у IPv4-адреса трех частей (сети, подсети и хоста), определенных маской *и классом A, B или C*.

ВНИМАНИЕ!

Контекст сертификации CCNA включает два других раздела, которые (к сожалению) также носят название *бесклассовый* и *классовый*. Кроме бесклассовой и классовой адресации, описанной здесь, существуют термины *бесклассовая маршрутизация* (classless routing) и *классовая маршрутизация* (classful routing), относящиеся к некоторым подробностям перенаправления маршрутизаторами Cisco пакетов с использованием стандартного маршрута. Кроме того, каждый протокол маршрутизации может быть отнесен к *бесклассовым протоколам маршрутизации* (classless routing protocol) или к *классовым протоколам маршрутизации* (classful routing protocol). В результате эти термины можно легко перепутать и неправильно использовать. Поэтому, когда видите слова, *бесклассовые* и *классовые*, обратите внимание на контекст их использования: адресация, маршрутизация или протоколы маршрутизации.

Выводы на основании формата IPv4-адреса

Зная, как разделить адрес, используя бесклассовые и классовые правила адресации, с помощью нескольких простых математических формул можно легко установить некоторые важные факты.

Сначала, зная количество битов хоста, для любой подсети можно вычислить количество IP-адресов хоста в подсети. Затем, если известно количество битов подсети (используется концепция классовой адресации) и то, что все подсети сети используют только одну маску, можно также вычислить количество подсетей в сети. Формулы требуют знания степеней числа 2.

Хостов в подсети. $2^H - 2$, где H — количество битов хоста.

Подсетей в сети. 2^S, где S — количество битов подсети. Используйте эту формулу, только если во всей сети применяется одна маска.

ВНИМАНИЕ!

В главе 11 приведено много подробностей о концепциях, связанных с масками, включая комментарии об одной маске для всей сети класса A, B или C.

Размеры частей IPv4-адресов также могут быть вычислены. Математика проста, но важны концепции. Имея в виду, что IPv4-адреса имеют длину 32 бита, обе части в бесклассовой адресации должны составить в целом 32 (P + H = 32), и в классовой адресации эти три части должны составить в целом 32 (N + S + H = 32). Эти отношения представлены на рис. 13.8.

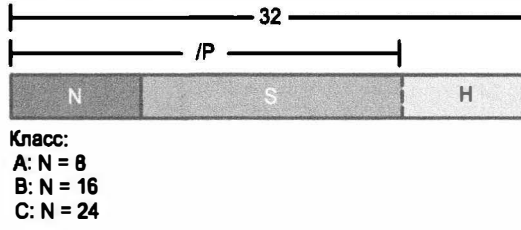


Рис. 13.8. Отношения между /P, N, S и H

При ответе на вопросы экзаменов CCENT и CCNA, а также при исследовании проблем в реальных сетях зачастую начинают с IP-адреса и маски. На основании

информации этой главы и предыдущих глав вполне можно найти всю информацию на рис. 13.8, а затем вычислить количество хостов на подсеть и количество подсетей в сети. Для справки этот процесс изложен поэтапно.

Ключевая тема **Формальные этапы анализа и вычисления значений, обсуждаемых в данной главе**

Этап 1 Преобразуйте маску в префиксный формат (/P), если нужно (см. выше)

Этап 2 На основании класса определите N (см. главу 12)

Этап 3 Вычислите S = P – N.

Этап 4 Вычислите H = 32 – P.

Этап 5 Вычислите количество хостов на подсеть: $2^H - 2$

Этап 6 Вычислите количество подсетей: 2^S

Рассмотрим, например, случай IP-адреса 8.1.4.5 с маской 255.255.0.0. Результат приведен ниже.

Этап 1 255.255.0.0 = /16, таким образом P=16

Этап 2 8.1.4.5 находится в диапазоне 1–126 первого октета, таким образом, это класс A, значит, N=8

Этап 3 S = P – N = 16 – 8 = 8.

Этап 4 H = 32 – P = 32 – 16 = 16.

Этап 5 $2^{16} - 2$ = 65 534 хоста на подсеть

Этап 6 2^8 = 256 подсетей

Анализ этой задачи представлен на рис. 13.9.

Рис. 13.9. Визуальное представление задачи: 8.1.4.5, 255.255.0.0

Для другого примера рассмотрим адрес 200.1.1.1 и маску 255.255.255.252. Результат приведен ниже.

Этап 1 255.255.255.252 = /30, таким образом, P=30

Этап 2 200.1.1.1 находится в диапазоне 192–223 первого октета, таким образом, это класс C, значит, N=24

Этап 3 S = P – N = 30 – 24 = 6.

Этап 4 H = 32 – P = 32 – 30 = 2.

Этап 5 $2^2 - 2$ = 2 хоста на подсеть

Этап 6 2^6 = 64 подсети

Этот пример использует популярную маску для последовательных каналов, поскольку последовательные каналы требуют только два адреса хоста и маску, поддерживающую только два адреса хоста.

Практические задания по анализу масок подсети

Прежде чем перейти к следующей главе, попрактикуйтесь в процессах, обсуждаемых в данной главе, пока не станете получать правильные ответы почти всегда. Используйте любые средства по своему усмотрению и любое время. Затем можно продолжить чтение.

Перед сдачей экзамена потренируйтесь, чтобы овладеть темой данной главы полностью и отвечать достаточно быстро. Что касается времени, ответ (размер трех частей плюс формулы для вычисления количества подсетей и хостов) нужно давать приблизительно через 15 секунд. Ключевые концепции и рекомендации такого двухэтапного подхода приведены в табл. 13.8.

Таблица 13.8. Продолжайте читать с учетом целей экзамена по темам данной главы

Период	Перед переходом к следующей главе	Перед сдачей экзамена
Сосредоточиться на ...	теме изучения	Быть быстрым и правильным
Разрешенные средства	Все	Ваш мозг и блокнот
Цель: точность	90% правильных ответов	100% правильных ответов
Цель: скорость	Любая скорость	15 секунд

Практические задания этой главы

На листе бумаги ответьте на следующие вопросы. В каждом случае:

- определите структуру адресов в каждой подсети на основании класса и маски, используя классовые концепции IP-адресации (другими словами, найдите размер частей сети, подсети и хоста адресов);
- вычислите количество хостов в подсети;
- вычислите количество подсетей в сети с учетом того, что повсюду используется та же маска.

1. 8.1.4.5, 255.255.254.0
2. 130.4.102.1, 255.255.255.0
3. 199.1.1.100, 255.255.255.0
4. 130.4.102.1, 255.255.252.0
5. 199.1.1.100, 255.255.255.224

Ответы приведены в разделе "Ответы на приведенные ранее практические задания".

Дополнительные практические задания

Для дополнительной практики по анализу масок подсети можно использовать следующее.

- Приложение Д на веб-сайте, в котором содержатся дополнительные практические задания. Оно содержит также объяснения по поиску ответа каждого задания.

- Приложение E на веб-сайте, которое содержит еще 25 практических заданий, связанных с этой главой. Хотя Приложение Д сосредоточено на темах данной главы, задачи в приложениях Д и E начинаются с IP-адреса и маски. Так, приложение E включает также комментарий и ответы на вопросы по количеству битов сети, подсети и хоста, а также другие темы, связанные с данной главой.
- Создайте собственные задания. Большинство калькуляторов подсети вычисляют количество битов сети, подсети и хоста, когда вводится IP-адрес и маска. Поэтому запишите IP-адрес и маску на бумаге, а затем найдите N, S и H. Затем, чтобы проверить ответ, используйте любой калькулятор подсети. Большинство калькуляторов подсети вычисляют класс и идентификатор сети. (Несколько калькуляторов предложено на веб-странице автора этой книги, указанной во введении.)

Обзор

Резюме

- Маски подсети могут быть записаны как 32-разрядные двоичные числа, но не любые. В частности, двоичная маска подсети должна следовать таким правилам:
  - значение не должно чередовать единицы и нули;
  - если есть единицы, они располагаются слева;
  - если есть нули, они располагаются справа.
- Преобразование между двоичным и префиксным форматами маски должно быть относительно интуитивно понятным, поскольку известно, что префиксное значение — это просто количество двоичных единиц в двоичной маске. Для законченности изложения рассмотрим преобразование в каждом направлении.
  - Из двоичной в префиксную. Подсчитайте количество единиц в двоичной маске и запишите его в десятичной форме после /.
  - Из префиксной в двоичную. Напишите количество единиц, соответствующее префиксному значению, и дополните их нулями до размера 32-разрядного двоичного числа.
- По определению десятичное число с разделительными точками (DDN), используемое в IPv4-адресации, содержит четыре десятичных числа, отделенных точками. Каждое десятичное число представляет 8 битов. Так, одно число DDN представляет четыре десятичных числа, которые вместе представляют некое 32-разрядное двоичное число.
- Маска подсети используется для следующих целей.
  - Определяет размер префиксной части (сети и подсети) адресов подсети.
  - Определяет размер части хоста адресов подсети.
  - Применяется при вычислении количества хостов в подсети.
  - Позволяет сетевому инженеру выяснить подробности о проекте подсети (количество битов подсети и хоста).
  - Согласно определению используется при вычислении количества подсетей во всей классовой сети.
  - Применяется в двоичных вычислениях идентификатора и широковещательного адреса подсети.
- Маска подсети разделяет IP-адреса подсети на две части: префикса (или подсети) и хоста.
- Часть префикса идентифицирует адреса, которые располагаются в той же подсети, поскольку у всех IP-адресов в той же подсети одинаковое значение в префиксной части их адресов.
  - Часть префикса (подсети). Одинаковы во всех адресах той же подсети.
  - Часть хоста. Различны во всех адресах той же подсети.
- Бесклассовая адресация. Концепция наличия у IPv4-адреса двух частей (префиксной и хоста), определенных только маской, без учета класса (A, B или C).
- Классовая адресация — концепция наличия у IPv4-адреса трех частей (сети, подсети и хоста), определенных маской и классом A, B или C.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Какие из следующих ответов представляют префиксный (CIDR) эквивалент маски 255.255.254.0?
   А) /19
   Б) /20
   В) /23
   Г) /24
   Д) /25
2. Какие из следующих ответов представляют префиксный (CIDR) эквивалент маски 255.255.255.240?
   А) /26
   Б) /28
   В) /27
   Г) /30
   Д) /29
3. Какие из следующих ответов представляют десятичный (DDN) эквивалент маски /24?
   А) 255.255.240.0
   Б) 255.255.252.0
   В) 255.255.255.0
   Г) 255.255.255.192
   Д) 255.255.255.240
4. Какие из следующих ответов представляют десятичный (DDN) эквивалент маски /30?
   А) 255.255.255.192
   Б) 255.255.255.252
   В) 255.255.255.240
   Г) 255.255.254.0
   Д) 255.255.255.0
5. Работая в службе технической поддержки, вы, получив звонок, изучаете IP-адрес (10.55.66.77) и маску (255.255.255.0) компьютера пользователя. Размышляя с точки зрения классовой логики, вы определяете количество битов сети (N), подсети (S) и хоста (H). Что из приведенного ниже истинно в данном случае?
   А) N=12
   Б) S=12
   В) H=8
   Г) S=8
   Д) N=24
6. Работая в службе технической поддержки, вы, получив звонок, изучаете IP-адрес и маску (192.168.9.1/27) компьютера пользователя. Размышляя с точки зрения классовой логики, вы определяете количество битов сети (N), подсети (S) и хоста (H). Что из приведенного ниже истинно в данном случае?

А) N=24
Б) S=24
В) H=8
Г) H=7

7. Инженер обдумывает с точки зрения логики бесклассовой IP-адресации следующий IP-адрес и маску: 10.55.66.77, 255.255.255.0. Какие из следующих утверждений истинны? (Выберите два ответа.)
   А) Размер части сети составляет 8 битов.
   Б) Длина префикса составляет 24 бита.
   В) Длина префикса составляет 16 битов.
   Г) Размер части хоста составляет 8 битов.
8. Какое из следующих утверждений истинно с точки зрения бесклассовых концепций IP-адресации?
   А) Используется 128-битовый IP-адрес.
   Б) Применимы только для сетей класса A и B.
   В) Разделяет IP-адреса на части сети, подсети и хоста.
   Г) Игнорирует правила сетей класса A, B и C.
9. Какая из следующих масок при использовании в качестве единой маски в пределах сети класса B предоставила бы достаточно битов подсети для поддержки 100 подсетей? (Выберите два ответа.)
   А)/24
   Б) 255.255.255.252
   В)/20
   Г) 255.255.252.0

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. ниже.

Таблица 13.9. Ключевые темы главы 13

Элемент	Описание	Страница
Список	Правила для двоичных значений маски подсети	394
Список	Правила преобразования двоичных и префиксных форм маски	395
Табл. 13.3	Девять значений, возможных в одном октете маски подсети	396
Список	Правила преобразования между двоичной и DDN формами маски	397
Список	Некоторые функции маски подсети	400
Список	Сравнение IP-адресов в одной подсети	401
Рис. 13.4	Части префикса (подсети) и хоста, разделенные единицами и нулями маски	401
Рис. 13.6	Применение концепции класса для создания трех частей адреса	402
Список	Определения классовой и бесклассовой адресации	402
Список	Формальные этапы анализа и вычисления значений, обсуждаемых в данной главе	404

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

двоичная маска (binary mask), десятичное представление с разделительными точками (Dotted-Decimal Notation — DDN), десятичная маска (decimal mask), префиксная маска (prefix mask), маска наклонной черты (slash mask), маска CIDR (CIDR mask), классовая адресация (classful addressing), бесклассовая адресация (classless addressing)

Практика

Если это еще не сделано, попрактикуйтесь в вопросах выявления деталей классовой сети, описанных в данной главе. Рекомендации приведены в разделе "Практические задания по преобразованию масок подсети".

Ответы на приведенные ранее практические задания

Набор практических заданий см. в табл. 13.7, а ответы приведены в табл. 13.10.

Таблица 13.10. Ответы на практические задания табл. 13.7

Префиксная	Двоичная маска	Десятичная
/18	11111111 11111111 11000000 00000000	255.255.192.0
/30	11111111 11111111 11111111 11111100	255.255.255.252
/25	11111111 11111111 11111111 10000000	255.255.255.128
/16	11111111 11111111 00000000 00000000	255.255.0.0
/8	11111111 00000000 00000000 00000000	255.0.0.0
/22	11111111 11111111 11111100 00000000	255.255.252.0
/15	11111111 11111110 00000000 00000000	255.254.0.0
/27	11111111 11111111 11111111 11100000	255.255.255.224

В табл. 13.11 приведены ответы к практическим заданиям раздела "Практические задания по анализу масок подсети".

Таблица 13.11. Ответы на практические задания этой главы

	Задача	/P	Класс	N	S	H	2^S	$2^H - 2$
1	8.1.4.5 255.255.254.0	23	A	8	15	9	32 768	510
2	130.4.102.1 255.255.255.0	24	B	16	8	8	256	254
3	199.1.1.100 255.255.255.0	24	C	24	0	8	Нет	254
4	130.4.102.1 255.255.252.0	22	B	16	6	10	64	1022
5	199.1.1.100 255.255.255.224	27	C	24	3	5	8	30

Ниже приведено описание заданий.

1. 8.1.4.5, первый октет (8) находится в диапазоне 1–126, следовательно, это адрес класса A с 8 битами сети. Маска 255.255.254.0 преобразуется в /23, следовательно, P – N = 15 для 15 битов подсети. H находим при вычитании /P (23) из 32, т.е. 9 битов хоста.
2. 130.4.102.1, первый октет находится в диапазоне 128–191, следовательно, это адрес класса B с N = 16 битов. 255.255.255.0 преобразуется в /24, следовательно, количество битов подсети 24 – 16 = 8. При 24 битах префикса количество битов хоста составляет 32 – 24 = 8.
3. Третье задание преднамеренно демонстрирует случай, когда маска не создает часть подсети адреса. Первый октет адреса 199.1.1.100 находится между 192 и 223, а следовательно, это адрес класса C с 24 битами сети. Префиксная версия маски /24, следовательно, количество битов подсети 24 – 24 = 0. Количество битов хоста 32 минус длина префикса (24) дает в общей сложности 8 битов хоста. Таким образом, в данном случае используется стандартная маска, которая не создает ни битов подсети, ни самих подсетей.
4. Адрес тот же, что и во втором задании, 130.4.102.1, принадлежит сети класса B с N = 16 битам. Но это задание использует другую маску, 255.255.252.0, которая преобразуется в /22. Это дает количество битов подсети 22 – 16 = 6. При 22 битах префикса количество битов хоста составляет 32 – 22 = 10.
5. Адрес тот же, что и в третьем задании, 199.1.1.100, принадлежит сети класса C с N = 24 битам. Но это задание использует другую маску, 255.255.255.224, которая преобразуется в /27. Это дает количество битов подсети 27 – 24 = 3. При 27 битах префикса количество битов хоста составляет 32 – 27 = 5.

Ответы на контрольные вопросы:

1 В. **2** Б. **3** В. **4** Б. **5** В. **6** А. **7** Б и Г. **8** Г. **9** А и Б.

ГЛАВА 14

Анализ существующих подсетей

Нередко задача начинается с поиска IP-адреса и маски, используемой неким хостом. Затем, чтобы понять, как объединенная сеть направляет пакеты на этот хост, следует найти основные элементы информации о подсети, а именно:

- идентификатор подсети;
- широковещательный адрес подсети;
- диапазон пригодных для использования одноадресатных IP-адресов подсети.

В этой главе обсуждаются концепции и математический механизм, позволяющие взять известный IP-адрес и маску, а затем полностью описать подсеть, находя значения из этого списка. Такие задачи, вероятно, обеспечат наиболее важные навыки в области IP-адресации и создания подсетей в данной книге, поскольку они чаще всего встречаются при работе и диагностике реальных сетей.

В этой главе рассматриваются следующие экзаменационные темы

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Определение подсети

Подсеть IP — это подмножество классовой сети, созданное по выбору сетевого инженера. Но инженер не может выбрать произвольное подмножество адресов; он должен следовать определенным правилам, которые приведены ниже.

Определение ключевых номеров подсети

Ключевая тема

- Подсеть содержит набор последовательных номеров.
- Подсеть содержит 2^H номеров, где H — количество битов хоста, определенных маской подсети.
- Два специальных номера в диапазоне не могут использоваться как IP-адреса.
  - Первый номер (наименьший) — это *идентификатор подсети* (subnet ID).
  - Последний номер (наибольший) — это *широковещательный адрес подсети* (subnet broadcast address).
- Остальные адреса, значения которых находятся между идентификатором и широковещательным адресом подсети, используются как *одноадресатные IP-адреса* (unicast IP address).

В этом разделе приведен обзор и подробно описаны концепции идентификатора подсети, широковещательного адреса подсети и диапазона адресов подсети.

Пример с сетью 172.16.0.0 и четырьмя подсетями

Предположим, вы работаете в центре поддержки и получаете звонки от пользователей, у которых проблемы с компьютером. Пользователь сообщает свой IP-адрес и маску: 172.16.150.41, 255.255.192.0. Одна из первых и наиболее распространенных задач, которые приходится решать на основании данной информации, — это поиск идентификатора подсети, в которой располагается указанный адрес. (Идентификатор подсети иногда называют резидентской подсетью, поскольку IP-адрес существует или располагается в этой подсети.)

Прежде чем обратиться к математике, исследуйте маску (255.255.192.0) и классовую сеть (172.16.0.0). Из маски, на основании изложенного в главе 13, можно вывести структуру адресов в подсети, включая количество битов подсети и хоста. Такой анализ свидетельствует о том, что адрес содержит два бита подсети, а значит, возможны четыре (2^2) подсети. (Если эти концепции еще не до конца очевидны, обратитесь к главе 13.) Структура адреса приведена на рис. 14.1.

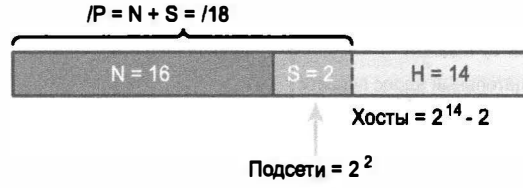


Рис. 14.1. Структура адреса: сеть класса B, маска /18

ВНИМАНИЕ!

В этой и других главах данной части подразумевается, что во всей классовой сети используется единая маска.

Поскольку каждая подсеть использует единую маску, все подсети данной сети IP имеют одинаковый размер, так как у всех подсетей одинаковая структура. В этом примере все четыре подсети имеют такую же структуру, как на рисунке, поэтому все они будут иметь $2^{14} - 2$ адреса хоста.

Рассмотрим общую картину подсети примера: у одной сети класса B теперь есть четыре подсети равного размера. Концептуально, если представить всю сеть класса B как числовую ось с четырьмя подсетями равного размера, каждая подсеть содержит, по существу, четверть сети, и каждая подсеть использует четверть номеров, как показано на рис. 14.2.. У каждой подсети есть идентификатор (самый маленький номер), поэтому на рисунке он находится слева, а также широковещательный адрес (самый большой номер), на рисунке он находится справа.

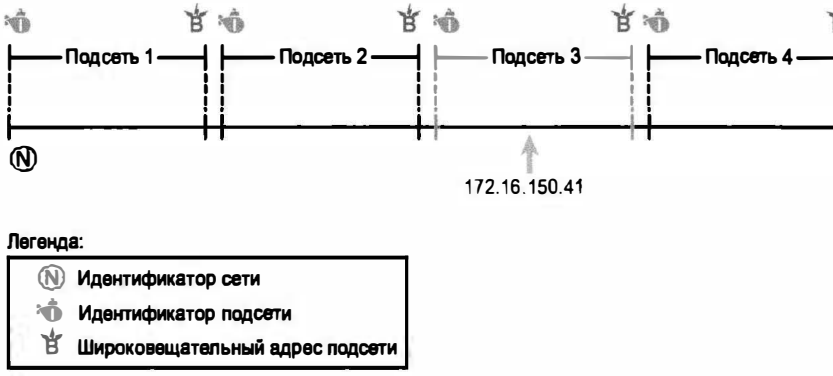


Рис. 14.2. Сеть 172.16.0.0, разделенная на четыре равных подсети

Остальная часть этой главы посвящена тому, как по IP-адресу и маске выяснить подробности той подсети, в которой он располагается. Другими словами, как найти резидентскую подсеть по IP-адресу. Снова используем IP-адрес 172.16.150.41 и маску 255.255.192.0. В примере на рис. 14.3 представлена резидентская подсеть наряду с идентификатором подсети и широковещательным адресом подсети, ограничивающими подсеть.

Рис. 14.3. Резидентская подсеть для 172.16.150.41, 255.255.192.0

Концепции идентификатора подсети

Идентификатор подсети — это число, используемое для ее краткого представления. Будучи указанным вместе с маской подсети, идентификатор подсети идентифицирует подсеть и применяется при получении широковещательного адреса, а также диапазона адресов подсети. Чтобы не записывать все эти подробности о подсети, достаточно записать идентификатор подсети и маску (их вполне достаточно для полного описания подсети).

Идентификатор подсети присутствует во многих местах, но чаще всего он используется в таблицах маршрутизации. Например, когда инженер задает маршрутизатору IP-адрес и маску, он вычисляет идентификатор подсети и помещает маршрут в свою таблицу маршрутизации для данной подсети. Затем, как правило, маршрутизатор анонсирует комбинацию идентификатор/маска подсети соседним маршрутизаторам с помощью некого протокола маршрутизации. В конечном счете все маршрутизаторы предприятия узнают о подсети (снова используя комбинацию идентификатора подсети и маски) и заносят ее в свои таблицы маршрутизации. (Содержимое таблицы маршрутизации маршрутизатора можно отобразить с помощью команды `show ip route`.)

К сожалению, терминология подсетей может иногда вызывать проблемы. Для начала скажем, что термины *идентификатор подсети* (subnet ID), *номер подсети* (subnet number) и *адрес подсети* (subnet address) — синонимы. Кроме того, люди иногда говорят просто *подсеть*, имея в виду и концепцию подсети, и число, используемое как идентификатор подсети. В разговоре о маршрутизации иногда используют термин *префикс* вместо термина *подсеть*. Термин *префикс* означает ту же идею, что и термин *подсеть*, но только используется в терминологии бесклассовой адресации как способ описания IP-адреса (см. главу 13).

Самый большой беспорядок в терминологии вызывают термины *сеть* (network) и *подсеть* (subnet). В реальном мире люди часто используют эти термины как синонимы, что в некоторых случаях совершенно резонно. В других случаях значения этих терминов специфичны, и различие между ними — вопрос обсуждаемой темы.

Например, зачастую спрашивают: "Каков идентификатор сети?", когда на самом деле хотят узнать идентификатор подсети. В другом случае речь могла бы идти об идентификаторе сети класса A, B или C. Поэтому когда инженер задает вопрос "Каков идентификатор сети 172.16.150.41/18?", используйте контекст, чтобы выяснить, нужен ли литерал идентификатора классовой сети (в данном случае 172.16.0.0) или литерал идентификатора подсети (в данном случае 172.16.128.0).

На экзамене следует быть готовым к этому и обращать внимание на контекст, когда используются термины *подсеть* и *сеть*, чтобы выяснить конкретное значение термина в данном случае.

Ключевые факты об идентификаторе подсети наряду с возможными синонимами приведены в табл. 14.1.

Таблица 14.1. Основные факты об идентификаторах подсетей

Ключевая тема

Определение	Число, представляющее подсеть
Числовое значение	Первое (наименьшее) число в подсети
Литеральные синонимы	Номер подсети, адрес подсети, префикс, резидентская подсеть
Обычное использование синонима	Сеть, идентификатор сети, номер сети, адрес сети
Обычно встречается в...	таблице маршрутизации, документации

Широковещательный адрес подсети

У широковещательного адреса подсети две основные роли: он используется как IP-адрес получателя при передаче пакетов всем хостам в подсети, а также при поиске старшего адреса в диапазоне допустимых адресов подсети.

Первоначально задача широковещательного адреса подсети состояла в том, чтобы предоставить хостам эффективный способ передачи одного пакета всем хостам в подсети. Например, хост в подсети A может послать пакет с адресом получателя, соответствующим широковещательному адресу подсети B. Маршрутизаторы перенаправят этот пакет точно так же, как пакет, посланный хосту в подсети B. Как только пакет достигнет последнего маршрутизатора, подключенного к подсети B, он перенаправит его всем хостам в подсети B, как правило, инкапсулируя пакет в широковещательный фрейм уровня управления передачей данных. В результате все хосты подсети B получат копию пакета.

Хотя широковещательный адрес подсети имеет небольшое практическое применение, на экзамене CCENT и CCNA он используется часто, поскольку широковещательный адрес — это последний (самый старший) адрес в диапазоне адресов подсети. Для поиска младшего конца диапазона вычислите идентификатор подсети, а для поиска старшего — широковещательный адрес подсети.

Ключевые факты о широковещательном адресе подсети наряду с возможными синонимами приведены в табл. 14.2.

Ключевая тема **Таблица 14.2. Основные факты о широковещательных адресах подсетей**

Частные сети IP	Класс сети
Определение	Зарезервированный адрес в каждой подсети, используемый как адрес получателя пакета, который заставляет маршрутизатор перенаправить пакет всем хостам в этой подсети
Числовое значение	Последнее (наибольшее) число в подсети
Литеральные синонимы	Направленный широковещательный адрес
Обычное использование синонима	Сетевое широковещание
Обычно встречается в...	вычислениях диапазона адресов подсети

Диапазон адресов, пригодных для использования

Инженер, реализующий объединенную сеть IP, должен знать диапазон одноадресатных IP-адресов в каждой подсети. Прежде чем можно будет запланировать, какие адреса использовать в качестве статически назначаемых IP-адресов при настройке сервера DHCP, а какие резервировать для дальнейшего использования, необходимо знать диапазон пригодных для использования адресов.

При поиске диапазона пригодных для использования IP-адресов в подсети найдите сначала идентификатор и широковещательный адрес подсети. Затем добавьте 1 к четвертому октету идентификатора подсети, чтобы получить первый (наименьший) пригодный для использования адрес, и вычтите 1 из четвертого октета широковещательного адреса подсети, чтобы получить последний (наибольший) пригодный для использования адрес в подсети.

Например, на рис. 14.3 представлены идентификатор подсети 172.16.128.0 и маска /18. Первый пригодный для использования адрес на единицу больше идентифи-

катора подсети (в данном случае 172.16.128.1). На том же рисунке широковещательный адрес подсети — 172.16.191.255, таким образом, последний пригодный для использования адрес на единицу меньше — 172.16.191.254.

Теперь, рассмотрев концепции чисел, которые совместно определяют подсеть, в остальной части этой главы сосредоточимся на математическом механизме, используемом для поиска этих значений.

Анализ существующих подсетей: двоичный

Что означает "проанализировать подсеть"? В данной книге это означает, что исходя из IP-адреса и маски нужно определить ключевые факты о подсети, в которой располагается этот адрес, а именно: выяснить идентификатор и широковещательный адрес подсети, а также диапазон адресов. Анализ может также включать вычисление количества адресов в подсети, как обсуждалось в главе 13, но в данной главе нет обзора этих концепций.

Есть много методов вычисления подробностей подсети на основании адреса и маски. Данный раздел начинается с обсуждения некоторых вычислений, в которых используется двоичная математика, а следующий раздел демонстрирует альтернативу — использование десятичной математики. Хотя для быстроты на экзаменах большинство людей предпочитают использовать десятичный метод, двоичные вычисления в конечном счете дают лучшее представление об IPv4-адресации. В частности, если планируется получить сертификат Cisco степенью выше CCNA, следует уделить время изучению двоичных методов, обсуждаемых в этом разделе, даже если для экзаменов используются десятичные методы.

Поиск идентификатора подсети: двоичный метод

В начале этого раздела, где используется двоичная математика, рассмотрим сначала простую десятичную математическую задачу: найти наименьшее десятичное трехзначное число, которое начинается с 4. Ответ, конечно, 400. Большинству людей, конечно, многоэтапная логика для этого не нужна, все знают, что 0 — самое малое значение, которое можно использовать для любой цифры в десятичном числе. Известно также, что первой цифрой должна быть 4, а количество знаков в числе — три, таким образом, используя самое низкое значение (0) для последних двух цифр, находят ответ: 400.

Эта же концепция применима для двоичных IP-адресов при вычислении идентификатора подсети. Подобные концепции уже были изложены в других главах, поэтому поиск идентификатора подсети в двоичном формате может быть интуитивно понятен! В противном случае следующие ключевые факты помогут понять логику.

- У всех адресов подсети (идентификатор и широковещательный адрес подсети и все пригодные для использования IP-адреса) одинаковое значение в префиксной части.
- Идентификатор подсети — наименьшее числовое значение в подсети, поэтому его часть хоста заполнена двоичными нулями.

Для поиска идентификатора подсети в двоичном формате возьмите IP-адрес в двоичном виде и замените все биты хоста на двоичные нули. Для этого необходимо преобразовать IP-адрес в двоичный формат. Необходимо также выявить биты префикса

и хоста, которые можно легко получить, преобразовав маску (по мере необходимости) в префиксный формат. (В приложении А содержится таблица десятично-двоичных преобразований.) На рис. 14.4 приведена концепция с использованием тех же адреса и маски, что и в прежних примерах этой главы: 172.16.150.41, маска /18.

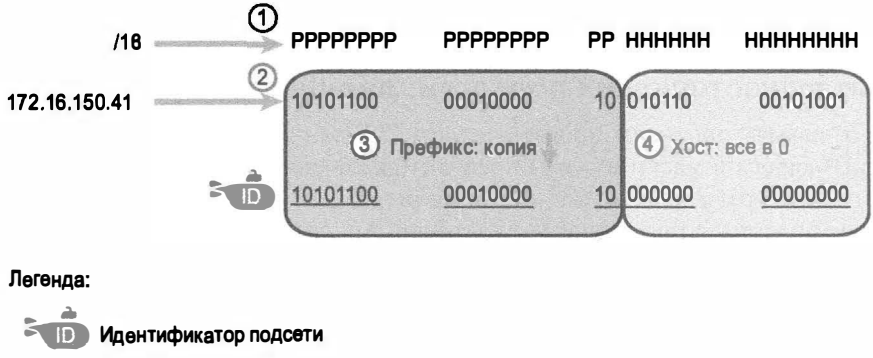


Рис. 14.4. Двоичная концепция: преобразование IP-адреса в идентификатор подсети

Начнем рассмотрение рис. 14.4 сверху: формат IP-адреса представлен в виде 18 битов префикса (P) и 14 битов хоста (H) маски (этап 1). Вторая строка (этап 2) демонстрирует двоичную версию IP-адреса, преобразованного из десятичного представления с разделительными точками (DDN) 172.16.150.41. (Если таблица преобразования из приложения А еще не использовалась, то самое время перепроверить преобразование всех четырех октетов на ее основании.)

Следующие два этапа демонстрируют действие: копирование битов префикса IP-адреса (этап 3) и присвоение битам хоста двоичных значений 0 (этап 4). В результате получается значение идентификатора подсети (в двоичном формате).

Последний этап, не представленный на рис. 14.4, подразумевает преобразование идентификатора подсети из двоичной системы счисления в десятичную. Здесь это преобразование представлено как отдельный этап на рис. 14.5 главным образом потому, что на данном этапе процесса многие делают ошибку. При преобразовании 32-разрядного числа (такого, как IP-адрес или идентификатор подсети) в формат DDN IPv4 необходимо придерживаться следующего правила:

> *преобразуйте из двоичной системы счисления в десятичную по 8 битов за раз, независимо от разделительной линии между частями префикса и хоста.*

Этот последний этап представлен на рис. 14.5. Обратите внимание на то, что у третьего октета (третий набор из 8 битов) два бита находятся в префиксе и шесть битов в части хоста, но преобразование происходит для всех восьми битов.

Поиск широковещательного адреса подсети: двоичный метод

При поиске широковещательного адреса подсети используется подобный процесс, но вместо записи всех битов части хоста наименьшим значением (бинарными нулями) они заполняются наибольшим значением (двоичными единицами). Данная концепция представлена на рис. 14.6.

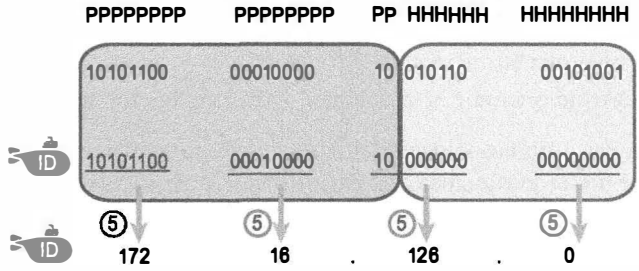


Рис. 14.5. Преобразование идентификатора подсети из двоичного формата в DDN

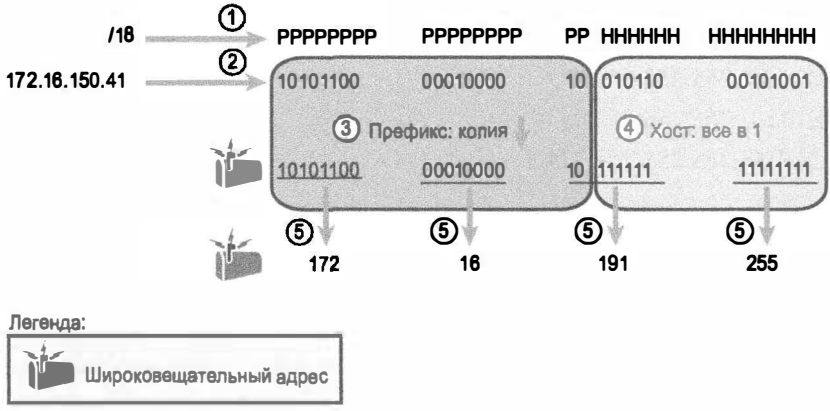


Рис. 14.6. Поиск широковещательного адреса подсети: двоичный метод

Первые три этапа процесса, приведенного на рис. 14.6, те же, что и на рис. 14.4. Это выявление битов префикса и хоста (этап 1), результат преобразования IP-адреса 172.16.150.41 в двоичный формат (этап 2) и копирование битов префикса (в данном случае первых 18 битов). Различие начинается в битах хоста справа — все они (в данном случае последние 14 битов) заменяются наибольшим возможным значением (двоичными единицами). На последнем этапе 32-разрядный широковещательный адрес подсети преобразуется в формат DDN. Кроме того, не забывайте, что при любом преобразовании из формата DDN в двоичный формат, и наоборот, используется по 8 битов за раз. В данном случае третий октет двоичных 10111111 преобразуется в десятичное значение 191.

Практические задания на применение двоичной математики

На рис. 14.4–14.6 приведен процесс поиска идентификатора подсети с использованием двоичной математики. Далее тот же процесс резюмирован в письменной форме для простоты изучения и применения.

Этапы применения двоичной математики для поиска идентификатора подсети Ключевая тема

Этап 1 Для нахождения длины префикса (/P) и длины части хоста (32 - P) преобразуйте маску в префиксный формат

Этап 2 Преобразуйте IP-адрес в его 32-разрядный двоичный эквивалент

Этап 3 Скопируйте префиксные биты IP-адреса

Этап 4 Запишите нули для битов хоста

Этап 5 Преобразуйте полученное 32-разрядное число, по 8 битов за раз, в десятичное число

Процесс поиска широковещательного адреса подсети совершенно такой же, кроме этапа 4, где биты записываются единицами, а не нулями (см. рис. 14.6).

Уделите некоторое время выполнению следующих пяти практических заданий на бумаге. В каждом случае найдите и идентификатор, и широковещательный адрес подсети. Кроме того, запишите маску в префиксном стиле.

1. 8.1.4.5, 255.255.0.0
2. 130.4.102.1, 255.255.255.0
3. 199.1.1.100, 255.255.255.0
4. 130.4.102.1, 255.255.252.0
5. 199.1.1.100, 255.255.255.224

В табл. 14.3–14.7 представлены результаты выполнения пяти заданий. Здесь биты хоста выделены полужирным шрифтом в двоичной версии адреса и маски, а также в двоичной версии идентификатора и широковещательного адреса подсети.

Таблица 14.3. Анализ подсети для адреса 8.1.4.5 и маски 255.255.0.0

Длина префикса	/16	11111111 11111111 **00000000 00000000**
Адрес	8.1.4.5	00001000 00000001 **00000100 00000101**
Идентификатор подсети	8.1.0.0	00001000 00000001 **00000000 00000000**
Широковещательный адрес	8.1.255.255	00001000 00000001 **11111111 11111111**

Таблица 14.4. Анализ подсети для адреса 130.4.102.1 и маски 255.255.255.0

Длина префикса	/24	11111111 11111111 11111111 **00000000**
Адрес	130.4.102.1	10000010 00000100 01100110 **00000001**
Идентификатор подсети	130.4.102.0	10000010 00000100 01100110 **00000000**
Широковещательный адрес	130.4.102.255	10000010 00000100 01100110 **11111111**

Таблица 14.5. Анализ подсети для адреса 199.1.1.100 и маски 255.255.255.0

Длина префикса	/24	11111111 11111111 11111111 **00000000**
Адрес	199.1.1.100	11000111 00000001 00000001 **01100100**
Идентификатор подсети	199.1.1.0	11000111 00000001 00000001 **00000000**
Широковещательный адрес	199.1.1.255	11000111 00000001 00000001 **11111111**

Таблица 14.6. Анализ подсети для адреса 130.4.102.1 и маски 255.255.252.0

Длина префикса	/22	11111111 11111111 111111**00 00000000**
Адрес	130.4.102.1	10000010 00000100 011001**10 00000001**
Идентификатор подсети	130.4.100.0	10000010 00000100 011001**00 00000000**
Широковещательный адрес	130.4.103.255	10000010 00000100 011001**11 11111111**

Таблица 14.7. Анализ подсети для адреса 199.1.1.100 и маски 255.255.255.224

Длина префикса	/27	11111111 11111111 11111111 111**00000**
Адрес	199.1.1.100	11000111 00000001 00000001 011**00100**
Идентификатор подсети	199.1.1.96	11000111 00000001 00000001 011**00000**
Широковещательный адрес	199.1.1.127	11000111 00000001 00000001 011**11111**

Сокращенный двоичный процесс

Описанный ранее в этом разделе двоичный процесс требует, чтобы все четыре октета были преобразованы в двоичные числа, а затем обратно в десятичные. Однако на основании маски DDN очень легко предсказать результаты по крайней мере трех из четырех октетов. Избежание двоичной математики во всех октетах, кроме одного, сокращает количество необходимых преобразований.

Сначала рассмотрим октет, значение маски DDN которого составляет 255. Десятичное значение 255 преобразуется в двоичное 11111111, значит, все 8 битов префиксные. На этапе 2 процесса адрес преобразуется в некое число. На этапе 3 число копируется. На этапе 4 то же 8-битовое число опять преобразуется в десятичное. Все сделанное на этих трех этапах в данном октете является преобразованием из десятичного числа в двоичное и преобразование того же числа в то же десятичное значение!

Короче говоря, идентификатор подсети (и широковещательный адрес подсети) равен IP-адресу в тех октетах, для которых маска содержит значение 255.

Рассмотрим, например, идентификатор 172.16.128.0 резидентской подсети 172.16.150.41 и маску 255.255.192.0. Первые два октета маски 255. Чтобы не применять двоичную математику, можно начать с копирования значения адреса в этих двух октетах: 172.16.

Для октетов, значение маски DDN которых составляет десятичный 0, сушествует другое сокращение. Десятичный 0 преобразуется в 8-битовое двоичное значение 00000000. Октет маски с 8 двоичными нулями означает, что все 8 битов в этом октете — биты хоста. И снова рассмотрим процесс из пяти этапов: на этапе 2 значение IP-адреса преобразуется в двоичное значение, но на этапе 4 все 8 этих битов преобразуются в 00000000, независимо от того, чем они были ранее. На этапе 5 этот двоичный октет 00000000 преобразуется назад в десятичное число, т.е. в десятичный 0. Таким образом, если некий октет маски DDN содержит десятичный 0, идентификатор подсети будет иметь десятичный 0 в том же октете и математических преобразований в этом октете можно избежать.

Следующие пересмотренные этапы процесса учитывают эти два сокращения. Однако, когда значение октета маски не равно ни 0, ни 255, процесс требует тех же преобразований, но максимум для одного октета. При поиске идентификатора подсети следующая логика применима для каждого из четырех октетов.

Общие этапы использования двоичной и десятичной математики для поиска идентификатора подсети

Ключевая тема

Этап 1 Если октет маски равен 255, копируйте десятичный октет IP-адреса

Этап 2 Если октет маски равен 0, запишите для него десятичный 0

Этап 3 Если октет маски не равен 0 и 255, используйте в этом октете ту же двоичную логику, что и в разделе "Поиск идентификатора подсети: двоичный метод"

Пример этого процесса приведен на рис. 14.7, опять же на примере 172.16.150.41, 255.255.192.0.

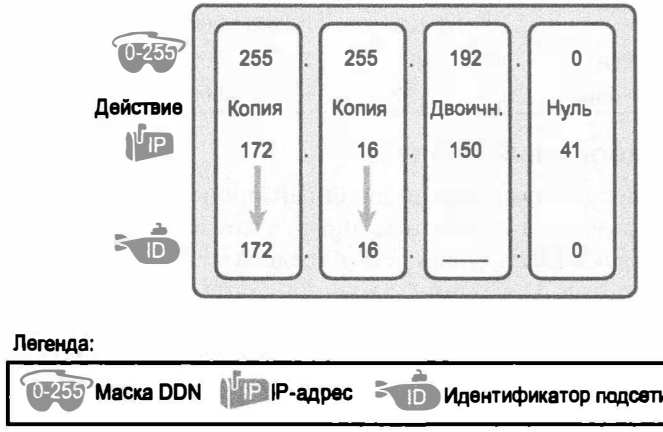


Рис. 14.7. Пример сокращения двоичного процесса

Подобное сокращенное вычисление существует и при поиске широковещательного адреса подсети. Для октетов маски DDN, равных десятичному 0, значение широковещательного адреса подсети принимается равным 255, а не 0, как указано в следующем списке.

Ключевая тема **Этапы использования двоичной и десятичной математики для поиска широковещательного адреса подсети**

Этап 1 Если октет маски равен 255, копируйте десятичный октет IP-адреса

Этап 2 Если октет маски равен 0, запишите для него десятичное 255

Этап 3 Если октет маски не равен 0 и 255, используйте в этом октете ту же двоичную логику, что и в разделе "Поиск широковещательного адреса подсети: двоичный метод"

Краткое замечание о логической математике

В этой главе было описано, как люди могут использовать двоичную математику для поиска идентификатора и широковещательного адреса подсети. Однако компьютеры используют для поиска тех же значений совершенно иной двоичный процесс на базе *Булевой алгебры*. Компьютеры уже хранят IP-адрес и маску в двоичном формате, поэтому они не должны делать никаких преобразований ни из десятичных чисел, ни в десятичные числа. Далее, операции Булевой алгебры позволяют компьютерам вычислять идентификатор и широковещательный адрес подсети всего за несколько действий процессора.

Не обязательно хорошо знать Булеву математику, чтобы иметь понятие о создании подсетей IP. Но если интересно, то вот как компьютеры используют Булеву логику для поиска идентификатора и широковещательного адреса подсети соответственно:

- выполнение булева И для IP-адреса и маски. Это преобразует все биты хоста в двоичные 0;
- инверсия маски и последующее булево ИЛИ для IP-адреса и инвертированной маски подсети. Это преобразует все биты хоста в двоичные 1.

Поиск диапазона адресов

Как только стали известны идентификатор и широковещательный адрес подсети, поиск диапазона пригодных для использования адресов подсети требует только простого сложения и вычитания. Для поиска первого (самого низкого) пригодного для использования IP-адреса в подсети добавьте 1 к четвертому октету идентификатора подсети. Для поиска последнего (самого высокого) пригодного для использования IP-адреса вычтите 1 из четвертого октета широковещательного адреса подсети.

Анализ существующих подсетей: десятичный

Анализ существующих подсетей с использованием двоичного процесса работает хорошо. Однако занимает у большинства людей много времени, особенно на десятично-двоичные преобразования, а на экзаменах Cisco CCENT и CCNA решать задачи следует быстро. На экзамене нужно быть в состоянии вычислить идентификатор подсети и диапазон пригодных для использования адресов по IP-адресу и маске в течение приблизительно 15 секунд. При использовании двоичных методов большинству людей требуется большой практический навык, чтобы даже с помощью сокращенного двоичного процесса найти эти ответы вовремя.

В данном разделе обсуждается, как находить идентификатор и широковещательный адрес подсети, используя только десятичную математику. Используя этот процесс, большинству людей проще и быстрее найти ответы, по крайней мере, после небольшой тренировки, по сравнению с двоичным процессом. Однако десятичный процесс ничего не скажет о смысле происходящего. Если вы не прочитали предыдущий раздел, "Анализ существующих подсетей: двоичный", имеет смысл прочитать его ради понимания процесса создания подсетей. Этот раздел посвящен быстрому получению правильного ответа с использованием подходящего метода.

Анализ с простыми масками

При трех простых масках поиск идентификатора и широковещательного адреса подсети требует лишь элементарной логики и почти никакой математики. Существуют три простых маски:

255.0.0.0
255.255.0.0
255.255.255.0

У этих масок есть только числа 255 и 0 в десятичном формате. По сравнению с ними у трудных масок есть один октет, значение отлично от 255 и 0, что усложняет логику.

ВНИМАНИЕ!

Термины *простая маска* (easy mask) и *трудная маска* (difficult mask) придуманы в этой книге для описания маски и уровня трудности при работе с ней.

Когда в задаче используется простая маска, можно достаточно быстро найти идентификатор подсети на основании IP-адреса и маски в формате DDN. Просто используйте при поиске идентификатора подсети следующий процесс для каждого из этих четырех октетов.

Этап 1 Если октет маски равен 255, скопируйте его десятичное значение из IP-адреса

Этап 2 Если октет маски равен 0, запишите десятичный 0

Для поиска широковещательного адреса подсети используйте подобный простой процесс следующим образом.

Этап 1 Если октет маски равен 255, скопируйте его десятичное значение из IP-адреса

Этап 2 Если октет маски равен 0, запишите десятичное 255

Прежде чем перейти к следующему разделу, уделите время заполнению табл. 14.8. Проверьте свои ответы по табл. 14.13 в разделе "Ответы на приведенные ранее практические задания" далее. Укажите в таблице идентификатор и широковещательный адрес подсети.

Таблица 14.8. Практическое задание: найдите идентификатор и широковещательный адрес подсети по простым маскам

	IP-адрес	Маска	Идентификатор подсети	Широковещательный адрес
1	10.77.55.3	255.255.255.0		
2	172.30.99.4	255.255.255.0		
3	192.168.6.54	255.255.255.0		
4	10.77.3.14	255.255.0.0		
5	172.22.55.77	255.255.0.0		
6	1.99.53.76	255.0.0.0		

Предсказуемость в интересующем октете

Хотя с тремя масками (255.0.0.0, 255.255.0.0 и 255.255.255.0) работать проще, остальные делают десятичную математику немного трудной, поэтому назовем эти маски трудными. У трудных масок значение одного октета отлично от 0 или 255. Математика в других трех октетах проста, а октет с более трудной математикой в книге называется *интересующим октетом*.

Если уделить время обдумыванию различных проблем и сосредоточиться на интересующем октете, то можно заметить определенный шаблон. Этот раздел демонстрирует, как использовать шаблон в десятичном числе и найти идентификатор подсети.

Значение идентификатора подсети имеет предсказуемое десятичное значение, если для всех подсетей одной классовой сети используется единая маска. Помните, в этой книге подразумевается, что для данной классовой сети разработчик решил использовать единую маску во всех подсетях. (Более подробную информацию по этой теме см. в главе 12.)

Для демонстрации предсказуемости рассмотрим рис. 14.8, на котором приведены некоторое идеи, учитываемые разработчиком при разделении сети на подсети. Здесь представлены четыре разных маски, рассматриваемые инженером для сети IPv4 класса B 172.16.0.0. На рисунке показаны значения третьего октета идентификаторов подсетей, которые были бы созданы при использовании масок 255.255.255.128, 255.255.255.192, 255.255.255.224 и 255.255.255.240 (сверху вниз на рисунке).

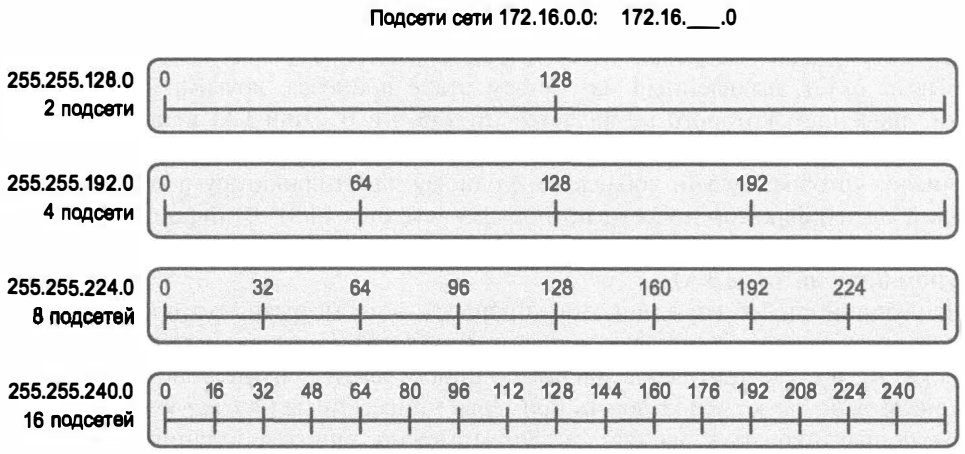


Рис. 14.8. Числовые шаблоны в интересующем октете

В первую очередь обратите внимание на верхнюю часть рисунка. Если инженер использует маску 255.255.255.128, получится две подсети с идентификаторами подсетей 172.16.0.0 и 172.16.128.0. Если инженер использует маску 255.255.255.192, получится четыре подсети с идентификаторами 172.16.0.0, 172.16.64.0.0, 172.16.128.0 и 172.16.192.0.

Шаблоны на рис. 14.8 очевидны. В данном случае:

маска: 255.255.255.128 — подсети кратны 128;
маска: 255.255.255.192 — подсети кратны 64;
маска: 255.255.255.224 — подсети кратны 32;
маска: 255.255.255.240 — подсети кратны 16.

Для поиска идентификатора подсети необходим только способ выяснить, каков шаблон. Если начинать приходится только с IP-адреса и маски, найдите идентификатор подсети как число, кратное магическому числу, ближайшее к IP-адресу без перекрытия (подробнее об этом — в следующем разделе).

Поиск идентификатора подсети: трудные маски

Ниже описаны все этапы процесса нахождения идентификатора подсети с использованием только десятичной математики. Этот процесс дополняет прежний процесс с использованием простых масок.

Этапы использования только десятичной математики для поиска идентификатора подсети

Ключевая тема

Этап 1 Если октет маски равен 255, скопируйте его десятичное значение из IP-адреса

Этап 2 Если октет маски равен 0, запишите десятичный 0

Этап 3 Если значение другое, считайте октет *интересующим*:

A. Вычислите *магическое число* как 256 – маска;

B. Установите значение идентификатора подсети как кратное магическому числу, ближайшее к IP-адресу без перекрытия

Процесс использует два новых термина, введенные в этой книге: *магическое число* (magic number) и *интересующий октет* (interesting octet). Термин *интересующий октет* описывает октет, выявленный на третьем этапе процесса; другими словами, октет маски, значением которого не является ни 255, ни 0. Этап 3 А) использует термин *магическое число*, происходящее от маски DDN. Концептуально магическое число — это число, которое, будучи добавлено к одному идентификатору подсети, даст следующий идентификатор подсети по порядку (см. рис. 14.8). В цифровой форме оно находится при вычитании значения маски DDN в интересующем октете, из числа 256, как упомянуто на этапе 3 А).

Наилучший способ изучения этого процесса — посмотреть, что происходит. Если можете, отложите пока книгу, смонтируйте образ DVD-диска, загруженный с веб-страницы книги, и посмотрите видеофильмы о поиске идентификатора подсети с трудной маской. Можно также использовать примеры, приведенные на следующих страницах, чтобы попрактиковаться на бумаге. Затем можно заняться заданиями из раздела "Практические задания по анализу существующих подсетей" далее в этой главе.

Резидентская подсеть (пример 1)

Рассмотрим, например, задание найти резидентскую подсеть для IP-адреса 130.4.102.1 и маски 255.255.240.0. Процесс не требует заботиться о битах префикса и хоста, преобразовании маски, рассмотрении маски в двоичном формате или преобразовании IP-адреса в двоичный формат и из него. Вместо этого для каждого из четырех октетов выберите действие на основании значения маски. На рис. 14.9 приведены результаты; номера в кружках соответствуют номерам этапов процесса поиска идентификатора подсети, приведенного выше в главе.

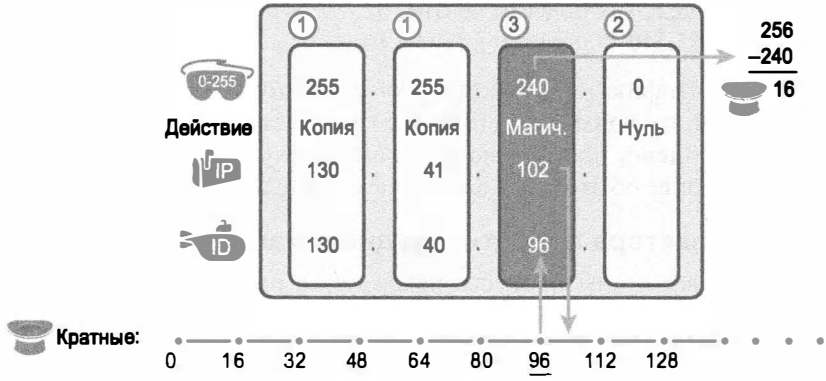


Рис. 14.9. Поиск идентификатора подсети: 130.4.102.1, 255.255.240.0

Сначала исследуйте три не интересующих октета (1, 2 и 4). Процесс основан на маске. У первых двух октетов маски есть значение 255, поэтому просто копируем октеты IP-адреса в октеты идентификатора подсети. У четвертого октета маски значение равно 0, поэтому для четвертого октета идентификатора подсети запишем 0.

Наиболее сложная логика находится в интересующем октете (третьем в данном случае) со значением маски 240. Для этого октета этап 3 А) подразумевает вычисление магического числа как 256 – маска. Это означает, что следует взять значение маски в интересующем октете (в данном случае 240) и вычесть его из 256: 256 – 240 = 16. Зна-

чения идентификатора подсети в этом октете должно быть кратно десятичному числу 16 в данном случае.

Далее, этап 3 B) подразумевает поиск значений, кратных магическому числу (в данном случае 16), и выбор самого близкого из них к IP-адресу без перекрытия. В частности, это означает, что необходимо мысленно вычислить значения, кратные магическому числу, начиная с 0. (Не забывайте 0!) Получаем набор чисел: 0, 16, 32, 48, 64, 80, 96, 112 и т.д. Затем найдите кратное значение, ближайшее к значению IP-адреса в этом октете (в данном случае 102), но не превышающее его. Как видно на рис. 14.9, это значение 96. Именно оно и выбирается для третьего октета идентификатора подсети 130.4.96.0.

Резидентская подсеть (пример 2)

Рассмотрим другой пример: 192.168.5.77 с маской 255.255.255.224. Результаты приведены на рис. 14.10.

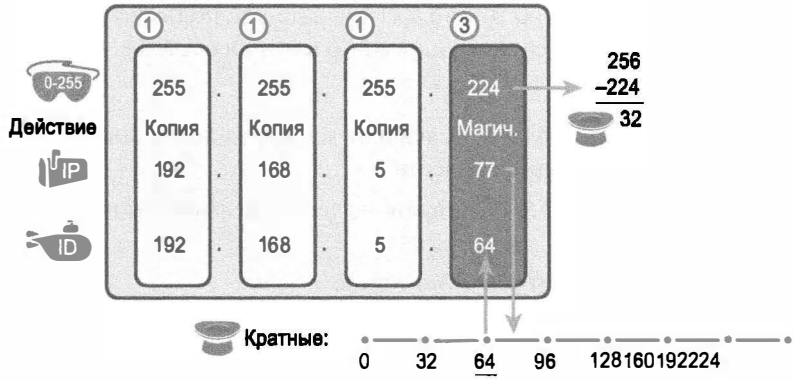


Рис. 14.10. Резидентская подсеть 192.168.5.77, 255.255.255.224

Три не интересующих октета (1, 2 и 3 в данном случае) требуют небольшого напряжения мысли. Для каждого октета со значением маски 255 достаточно скопировать значение из IP-адреса.

Для интересующего октета на этапе 3 A) магическое число составит 256 – 224 = 32. Кратные магическому числу числа: 0, 32, 64, 96 и т.д. Поскольку значением IP-адреса в четвертом октете является 77, кратное число должно быть ближайшим к нему, но без перекрытия; поэтому идентификатор подсети закончится на 64 (192.168.5.64).

Практические задания по резидентским подсетям

Прежде чем переходить к следующему разделу, уделите время заполнению табл. 14.9. Проверьте свои ответы по табл. 14.14 в разделе "Ответы на приведенные ранее практические задания", приведенном далее. Заполните в таблице столбец идентификатора подсети по каждому случаю. Объяснения выполнения каждого задания приведено после табл. 14.14.

Таблица 14.9. Практическое задание: поиск идентификатора подсети: трудные маски

Задание	IP-адрес	Маска	Идентификатор подсети
1	10.77.55.3	255.248.0.0	
2	172.30.99.4	255.255.192.0	
3	192.168.6.54	255.255.255.252	
4	10.77.3.14	255.255.128.0	
5	172.22.55.77	255.255.254.0	
6	1.99.53.76	255.255.255.248	

Поиск широковещательного адреса подсети: трудные маски

Для поиска широковещательного адреса подсети применяется подобный процесс. Для простоты он начинается с идентификатора подсети, а не с IP-адреса. Если вам случится начинать с IP-адреса, используйте описанные в этой главе процессы, чтобы сначала найти идентификатор подсети, а затем используйте следующий процесс для поиска широковещательного адреса подсети для той же подсети. Для каждого октета сделайте следующее.

Ключевая тема

Этапы использования только десятичной математики для поиска широковещательного адреса подсети

Этап 1 Если октет маски равен 255, скопируйте значение идентификатора подсети

Этап 2 Если октет маски равен 0, запишите 255

Этап 3 Если значение другое, считайте октет *интересующим*:

A. Вычислите *магическое число* как 256 – маска;

B. Возьмите значение идентификатора подсети, добавьте магическое число и вычтите 1 (*идентификатор* + *магическое* – *1*)

Подобно процессу, использованному при поиске идентификатора подсети, есть несколько возможностей для лучшего изучения и усвоения процесса. Сейчас можно отложить чтение и просмотреть видеофильм о поиске широковещательного адреса подсети с трудной маской. Кроме того, посмотрите примеры в этом разделе, которые демонстрируют этот процесс на бумаге. Затем обратитесь к заданиям из раздела "Дополнительные практические задания" данной главы.

Широковещательный адрес подсети (пример 1)

Данный пример является продолжением первого примера из раздела "Поиск идентификатора подсети: трудные маски", представленного на рис. 14.9. Этот пример начинается с IP-адреса и маски (130.4.102.1, 255.255.240.0) и демонстрирует поиск идентификатора подсети 130.4.96.0. Рис. 14.11 начинается теперь с идентификатора подсети и той же маски.

Сначала исследуйте три не интересующих октета (1, 2 и 4). Первые два октета маски имеют значение 255, поэтому просто копируем октет идентификатора подсети в соответствующий октет широковещательного адреса подсети. В четвертом октете маски значение 0, поэтому для четвертого октета запишите 255.

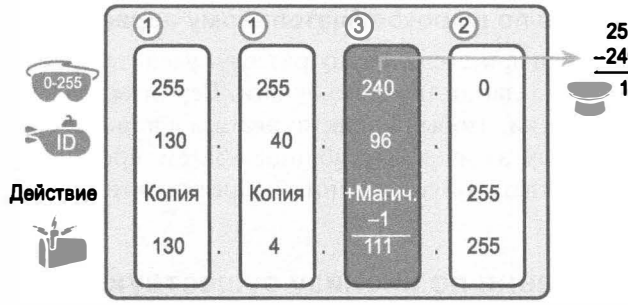


Рис. 14.11. Поиск широковещательного адреса подсети: 130.4.96.0, 255.255.240.0

Интересующим октетом в данном примере является третий, из-за значения 240 маски. Сначала, на этапе 3 А), необходимо вычислить магическое число как 256 – маска. (Если идентификатор подсети уже вычислен в ходе приведенного ранее процесса, магическое число должно быть известно.) На этапе 3 В) добавьте к значению идентификатора подсети (96) магическое число (16) и вычтите 1, в результате получится 111. Это даст широковещательный адрес подсети 130.4.111.255.

Широковещательный адрес подсети (пример 2)

Этот пример также продолжает предыдущий пример из раздела "Поиск идентификатора подсети: трудные маски", представленный на рис. 14.10. Пример демонстрировал, как, начиная с IP-адреса 192.168.5.77 и маски 255.255.255.224, найти идентификатор подсети 192.168.5.64. Рис. 14.12 начинается с этого идентификатора подсети и той же маски.

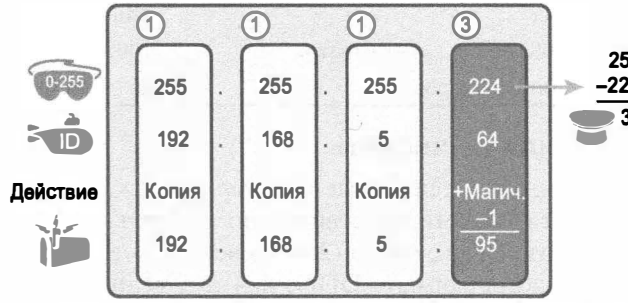


Рис. 14.12. Поиск широковещательного адреса подсети: 192.168.5.64, 255.255.255.224

Сначала исследуйте три не интересующих октета (1, 2 и 3). У первых трех октетов маски значение 255, поэтому просто копируем октет идентификатора подсети в соответствующий октет широковещательного адреса подсети.

Четвертый октет этого примера интересующий, поскольку имеет значение 224 маски. Сначала этап 3 А) требует вычислить магическое число как 256 – маска. (Если идентификатор подсети уже вычислялся, то это то же магическое число, поскольку используется та же маска.) На этапе 3 В) добавьте к значению идентификатора подсети (64) магическое значение (32) и вычтите 1, в результате получится 95. Это даст широковещательный адрес подсети 192.168.5.95.

Практические задания по широковещательному адресу подсети

Прежде чем переходить к следующему разделу, уделите время выполнению нескольких практических заданий на листе бумаги. Вернитесь к табл. 14.10, где перечислены IP-адреса и маски, чтобы попрактиковаться в поиске широковещательного адреса подсети для всех задач этой таблицы. Затем проверьте свои ответы по табл. 14.15 в разделе "Ответы на приведенные ранее практические задания", приведенном далее.

Практические задания по анализу существующих подсетей

Перед тем как перейти к следующей главе, попрактикуйтесь в процессах, обсуждаемых в данной главе, пока не станете почти всегда получать правильные ответы. Используйте любые средства по своему усмотрению и любое время. Затем можно продолжить чтение.

Перед сдачей экзамена потренируйтесь, чтобы овладеть темой данной главы полностью и отвечать достаточно быстро. Что касается времени, то идентификатор подсети на основании IP-адреса и маски следует найти приблизительно через 15 секунд. Следует также стремиться, начиная с идентификатора подсети и маски, находить широковещательный адрес и диапазон адресов еще через 10–15 секунд. Ключевые концепции и рекомендации такого двухэтапного подхода приведены в табл. 14.10.

Таблица 14.10. Продолжайте читать с учетом целей экзамена по темам данной главы

Период	Перед переходом к следующей главе	Перед сдачей экзамена
Сосредоточиться на ...	теме изучения	быть быстрым и правильным
Разрешенные средства	Все	Ваш мозг и блокнот
Цель: точность	90% правильных ответов	100% правильных ответов
Цель: скорость	Любая скорость	20–30 секунд

Выбор: запомнить или вычислять

Как описано в этой главе, десятичные процессы поиска идентификатора и широковещательного адреса подсети действительно требуют некоторых вычислений, включая вычисление магического числа (256 – маска). Эти же процессы подразумевают использование маски в формате DDN, поэтому если в экзаменационном вопросе маска дана в префиксном стиле, то перед использованием описанного здесь процесса ее необходимо преобразовать в формат DDN.

За эти годы многие люди говорили мне, что предпочитают запоминать таблицу для поиска магического числа. В этой таблице перечислены магические числа для разных масок и префиксные маски, таким образом, вы избегаете преобразования из префиксного формата маски в DDN. Табл. 14.11 — пример такой таблицы. Не стесняйтесь игнорировать или использовать данную таблицу — это сугубо ваш выбор.

Таблица 14.11. Справочная таблица: значения маски DDN, двоичный эквивалент, магические числа и префиксы

Префикс, интересующий октет 2	/9	/10	/11	/12	/13	/14	/15	/16
Префикс, интересующий октет 3	/17	/18	/19	/20	/21	/22	/23	/24
Префикс, интересующий октет 4	/25	/26	/27	/28	/29	/30		
Магическое число	128	64	32	16	8	4	2	1
Маска DDN в интересующем октете	128	192	224	240	248	252	254	255

Дополнительные практические задания

В этом разделе перечислены некоторые возможности для дополнительной практики.

- Приложение E на веб-сайте, в котором содержатся дополнительные практические задания, а также объяснения по поиску ответа на каждое задание.
- Создайте собственные задания. Большинство калькуляторов подсети позволяют найти биты сети, подсети и хоста, когда вводят IP-адрес и маску, поэтому запишите IP-адрес и маску на бумаге, а затем найдите идентификатор подсети и диапазон адресов. Затем, чтобы проверить результат, используйте любой калькулятор подсети. (Несколько калькуляторов предложено на веб-странице автора этой книги, указанной во введении.)

Обзор

Резюме

- Подсеть IP — это подмножество классовой сети, созданное по выбору сетевого инженера. Но инженер не может выбрать произвольное подмножество адресов; он должен следовать определенным правилам, которые приведены ниже.
  - Подсеть содержит набор последовательных номеров.
  - Подсеть содержит 2^H номеров, где H — количество битов хоста, определенных маской подсети.
  - Два специальных номера в диапазоне не могут использоваться как IP-адреса.
  - Первый номер (наименьший) — это идентификатор подсети.
  - Последний номер (наибольший) — это широковещательный адрес подсети.
- Остальные адреса, значения которых находятся между идентификатором и широковещательным адресом подсети.
- Идентификатор подсети — это число, используемое для ее краткого представления. Будучи указанным вместе с маской подсети, идентификатор подсети идентифицирует подсеть и применяется при получении широковещательного адреса, а также диапазона адресов подсети. Чтобы не записывать все эти подробности о подсети, достаточно записать идентификатор подсети и маску (их вполне достаточно для полного описания подсети).
- Идентификатор подсети присутствует во многих местах, но чаще всего используется в таблицах маршрутизации.
- Термины *идентификатор подсети*, *номер подсети* и *адрес подсети* — синонимы.
- У широковещательного адреса подсети две основные роли: он используется как IP-адрес получателя при передаче пакетов всем хостам в подсети, а также при поиске старшего адреса в диапазоне допустимых адресов подсети.
- При поиске диапазона пригодных для использования IP-адресов в подсети найдите сначала идентификатор и широковещательный адрес подсети. Затем добавьте 1 к четвертому октету идентификатора подсети, чтобы получить первый (наименьший) пригодный для использования адрес, и вычтите 1 из четвертого октета широковещательного адреса подсети, чтобы получить последний (наибольший) пригодный для использования адрес в подсети.
- У всех адресов подсети (идентификатор подсети, широковещательный адрес подсети и все пригодные для использования IP-адреса) одинаковое значение в префиксной части.
- Идентификатор подсети — самое низкое числовое значение в подсети, поэтому его часть хоста заполнена двоичными нулями.
- Процесс поиска идентификатора подсети с использованием двоичной математики:

Этап 1 Для нахождения длины префикса (/P) и длины части хоста (32 - P) преобразуйте маску в префиксный формат

Этап 2 Преобразуйте IP-адрес в его 32-разрядный двоичный эквивалент

Этап 3 Скопируйте префиксные биты IP-адреса

Этап 4 Запишите нули для битов хоста

Этап 5 Преобразуйте полученное 32-разрядное число, по 8 битов за раз, в десятичное число

Процесс поиска широковещательного адреса подсети совершенно такой же, кроме этапа 4, где биты записываются единицами, а не нулями.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. С точки зрения правил классовой адресации у IP-адреса может быть три части: сети, подсети и хоста. Если исследовать все адреса в одной подсети в двоичном формате, какое из следующих утверждений правильно описывает совпадение частей всех адресов? Выберите лучший ответ.

   А) Только части сети.

   Б) Только части подсети.

   В) Только части хоста.

   Г) Части сети и подсети.

   Д) Части подсети и хоста.

2. Какое из следующих утверждений истинно исходя из двоичных значений идентификатора подсети, широковещательного адреса подсети и IP-адреса хоста в какой-нибудь одной подсети? (Выберите два ответа.)

   А) Вся часть хоста широковещательного адреса состоит из двоичных нулей.

   Б) Вся часть хоста идентификатора подсети состоит из двоичных нулей.

   В) Вся часть хоста пригодного для использования IP-адреса может состоять из двоичных единиц.

   Г) Часть хоста любого пригодного для использования IP-адреса не должна состоять исключительно из двоичных нулей.

3. Что из следующего является резидентским идентификатором подсети для IP-адреса 10.7.99.133/24?

   А) 10.0.0.0

   Б) 10.7.0.0

   В) 10.7.99.0

   Г) 10.7.99.128

4. Что из следующего является резидентской подсетью для IP-адреса 192.168.44.97/30?

   А) 192.168.44.0

   Б) 192.168.44.64

   В) 192.168.44.96

   Г) 192.168.44.128

5. Что из следующего является широковещательным адресом подсети, в которой располагается IP-адрес 172.31.77.201/27?

   А) 172.31.201.255
   Б) 172.31.255.255
   В) 172.31.77.223
   Г) 172.31.77.207

6. Некий инженер просит вас настроить сервер DHCP так, чтобы зарезервировать 100 последних пригодных для использования IP-адресов в подсети 10.1.4.0/23. Какой из следующих IP-адресов мог бы оказаться зарезервированным в результате новой конфигурации?

   А) 10.1.4.156
   Б) 10.1.4.254
   В) 10.1.5.200
   Г) 10.1.7.200
   Д) 10.1.255.200

7. Некий инженер просит вас настроить сервер DHCP так, чтобы зарезервировать 20 первых пригодных для использования IP-адресов в подсети 192.168.9.96/27. Какой из следующих IP-адресов мог бы оказаться зарезервированным в результате новой конфигурации?

   А) 192.168.9.126
   Б) 192.168.9.110
   В) 192.168.9.1
   Г) 192.168.9.119

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 14.2.

Таблица 14.12. Ключевые темы главы 14

Элемент	Описание	Страница
Список	Определение ключевых номеров подсети	413
Табл. 14.1	Основные факты об идентификаторах подсетей	415
Табл. 14.2	Основные факты о широковещательных адресах подсетей	416
Список	Этапы применения двоичной математики для поиска идентификатора подсети	419
Список	Общие этапы использования двоичной и десятичной математики для поиска идентификатора подсети	421
Список	Этапы использования двоичной и десятичной математики для поиска широковещательного адреса подсети	422
Список	Этапы использования только десятичной математики для поиска идентификатора подсети	425
Список	Этапы использования только десятичной математики для поиска широковещательного адреса подсети	428

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

резидентская подсеть (resident subnet), идентификатор подсети (subnet ID), номер подсети (subnet number), адрес подсети (subnet address), широковещательный адрес подсети (subnet broadcast address)

Практика

Если это еще не сделано, попрактикуйтесь в вопросах поиска идентификатора подсети, диапазона адресов и широковещательного адреса подсети, связанных с IP-адресом и маской, описанных в данной главе. Рекомендации приведены в разделе "Практические задания по анализу существующих подсетей".

Ответы на приведенные ранее практические задания

Эта глава содержит множество заданий, распределенных по всей главе. Ответы приведены в табл. 14.13–14.15.

Таблица 14.13. Ответы на практические задания табл. 14.8

	IP-адрес	Маска	Идентификатор подсети	Широковещательный адрес
1	10.77.55.3	255.255.255.0	10.77.55.0	10.77.55.255
2	172.30.99.4	255.255.255.0	172.30.99.0	172.30.99.255
3	192.168.6.54	255.255.255.0	192.168.6.0	192.168.6.255
4	10.77.3.14	255.255.0.0	10.77.0.0	10.77.255.255
5	172.22.55.77	255.255.0.0	172.22.0.0	172.22.255.255
6	1.99.53.76	255.0.0.0	1.0.0.0	1.255.255.255

Таблица 14.14. Ответы на практические задания табл. 14.9

	IP-адрес	Маска	Идентификатор подсети
1	10.77.55.3	255.248.0.0	10.72.0.0
2	172.30.99.4	255.255.192.0	172.30.64.0
3	192.168.6.54	255.255.255.252	192.168.6.52
4	10.77.3.14	255.255.128.0	10.77.0.0
5	172.22.55.77	255.255.254.0	172.22.54.0
6	1.99.53.76	255.255.255.248	1.99.53.72

Ниже приведены объяснения ответов для табл. 14.14.

1. Интересующий октет — второй, с магическим числом 256 – 248 = 8. К числам, кратным 8, относятся: 0, 8, 16, 24, ..., 64, 72 и 80. Число 72 является самым близким к значению IP-адреса в том же октете (77) без превышения. В результате получаем идентификатор подсети 10.72.0.0.
2. Интересующий октет — третий, с магическим числом 256 – 192 = 64. К числам, кратным 64, относятся: 0, 64, 128 и 192. Число 64 является самым близким к значению IP-адреса в том же октете (99) без превышения. В результате получаем идентификатор подсети 172.30.64.0.
3. Интересующий октет четвертый, с магическим числом 256 – 252 = 4. К числам, кратным 4, относятся: 0, 4, 8, 12, 16, ..., 48, 52 и 56. Число 52 является самым близким к значению IP-адреса в том же октете (54) без превышения. В результате получаем идентификатор подсети 192.168.6.52.
4. Интересующий октет третий, с магическим числом 256 – 128 = 128. Для этого случая существуют только два кратных значения: 0 и 128. Число 0 является самым близким к значению IP-адреса в том же октете (3) без превышения. В результате получаем идентификатор подсети 10.77.0.0.
5. Интересующий октет — третий, с магическим числом 256 – 254 = 2. К числам, кратным 2, относятся: 0, 2, 4, 6, 8 и так далее, по существу, все четные числа. Число 54 является самым близким к значению IP-адреса в том же октете (55) без превышения. В результате получаем идентификатор подсети 172.22.54.0.
6. Интересующий октет — четвертый, с магическим числом 256 – 248 = 8. К числам, кратным 8, относятся: 0, 8, 16, 24, ..., 64, 72 и 80. Число 72 является самым близким к значению IP-адреса в том же октете (76) без превышения. В результате получаем идентификатор подсети 1.99.53.72.

Таблица 14.15. Ответы на практические задания раздела "Практические задания по широковещательному адресу подсети"

	Идентификатор подсети	Маска	Широковещательный адрес
1	10.72.0.0	255.248.0.0	10.79.255.255
2	172.30.64.0	255.255.192.0	172.30.127.255
3	192.168.6.52	255.255.255.252	192.168.6.55
4	10.77.0.0	255.255.128.0	10.77.127.255
5	172.22.54.0	255.255.254.0	172.22.55.255
6	1.99.53.72	255.255.255.248	1.99.53.79

Ниже приведены объяснения ответов для табл. 14.15.

1. Интересующий октет — второй. Обработка трех простых октетов дает широковещательный адрес без интересуюшего октета 10.___.255.255. С магическим числом 256 – 248 = 8, второй октет будет 72 (из идентификатора подсети) плюс 8 минус 1, или 79.

2. Интересующий октет — третий. Обработка трех простых октетов дает широковещательный адрес без интересующего октета 172.30.___.255. С магическим числом 256 – 192 = 64 интересующий октет будет 64 (из идентификатора подсети) плюс 64 (магическое число) минус 1, или 127.
3. Интересующий октет — четвертый. Обработка трех простых октетов дает широковещательный адрес без интересующего октета 192.168.6.___. С магическим числом 256 – 252 = 4 интересующий октет будет 52 (значение идентификатора подсети) плюс 4 (магическое число) минус 1, или 55.
4. Интересующий октет — третий. Обработка трех простых октетов дает широковещательный адрес без интересующего октета 10.77.___.255. С магическим числом 256 – 128 = 128 интересующий октет будет 0 (значение идентификатора подсети) плюс 128 (магическое число) минус 1, или 127.
5. Интересующий октет — третий. Обработка трех простых октетов дает широковещательный адрес без интересующего октета 172.22.___.255. С магическим числом 256 – 254 = 2 широковещательный адрес в интересующем октете будет 54 (значение идентификатора подсети) плюс 2 (магическое число) минус 1, или 55.
6. Интересующий октет — четвертый. Обработка трех простых октетов дает широковещательный адрес без интересующего октета 1.99.53.___. С магическим числом 256 – 248 = 8 широковещательный адрес в интересующем октете будет 72 (значение идентификатора подсети) плюс 8 (магическое число) минус 1, или 79.

Ответы на контрольные вопросы:

1 Г. **2** Б и Г. **3** В. **4** В. **5** В. **6** В. **7** Б.

Обзор части III

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части III

Задача	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей терминов подсети		
Создайте диаграмму связей вычислений подсети		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав только этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения к книге.

Ответы на вопросы

Ответьте на вопросы обзора этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров только этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения к книге.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если вам не все понятно, уделите время повторному изучению.

Создайте диаграмму связей терминов подсети

Тема IPv4-адресации и создания подсетей имеет много терминов, многие из которых являются синонимами, некоторые имеют подобные значения или иной смысл в другом контексте. Первая диаграмма связей обзора части III требует организовать все термины IP-адресации и создания подсетей, которые вы помните, по четырем разделам, и в каждом разделе распределить термины согласно тому, являются ли они или синонимами, подобными терминами или описанием.

Чтобы дать общее представление, на рис. Ч3.1 приведен пример начала одной ветви диаграммы связей. Для этой ветви достаточно вспомнить все термины, связанные с IP address (IP-адресом), и разместить их в одном из этих трех разделов. Ваша диаграмма, конечно, может выглядеть иначе. Как обычно, выполните это задание сначала без книги и своих заметок. Затем, воспользовавшись книгой, удостоверьтесь, что включили, по крайней мере, все термины из раздела ключевых терминов в конце глав.

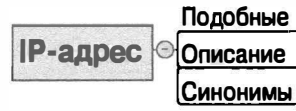


Рис. Ч3.1. Пример начального этапа первой диаграммы связей части III

ВНИМАНИЕ!

Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения к данной книге.

Создайте диаграмму связей вычислений подсети

В этой части рассматривается несколько типов задач на создание подсетей, которые следует проанализировать и решить. Следующее упражнение с диаграммой связей поможет сделать общий обзор задач всех типов. Этот обзор не сосредоточивается на подробностях поиска ответа на каждую задачу — для этого есть практические задания в конце глав 12–14. В этих главах рассматриваются четыре основные типа задач, которые могут быть решены арифметически.

- *Факты о классовых сетях.* Выясните на основании IP-адреса как можно больше фактов о его классовой сети IP.
- *Преобразование маски.* По заданной маске подсети найдите ее эквивалент в двух других форматах.
- *Анализ маски.* По заданным маске подсети и адресу найдите количество хостов на подсеть и количество подсетей.
- *Анализ подсети.* По заданным маске подсети и адресу найдите номера, определяющие резидентскую подсеть (идентификатор подсети, широковещательный адрес подсети и диапазон пригодных для использования IP-адресов).

Создайте диаграмму связей с четырьмя ветвями, по одной для каждой темы в списке. Начните каждую ветвь с базовой концепции и разбейте ее на три подраздела.

- *Дано* (given). Имеющаяся и подразумеваемая информация, на основании которой предстоит решить задачу.
- *Процесс* (process). Информация или термины, используемые во время процесса. Не описывайте отдельные этапы процесса (главное, чтобы вспомнить, что это именно этот процесс, а не другой).
- *Результат* (result). Факты, определяемые при решении задачи.

На рис. Ч3.2 приведен пример диаграммы (неполный) для фактов о классовых сетях, только чтобы продемонстрировать основную идею.

Если решено использовать программное обеспечение диаграмм связей, а не лист бумаги, имеет смысл запомнить место сохранения файлов диаграмм связей; в таблице ниже перечислены диаграммы связей для данной части, имена их файлов и места сохранения.

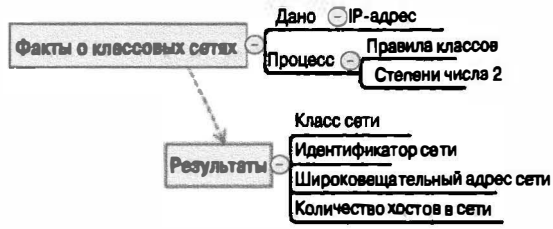


Рис. Ч3.2. Пример диаграммы связей для фактов о классовых сетях

Диаграммы связей обзора части III

Диаграмма	Описание	Где сохранен результат
1	Диаграмма терминов подсети	
2	Диаграмма вычислений подсети	

Ответы приведены в приложении П (Appendix O) на веб-сайте, но ваши диаграммы связей могут выглядеть иначе.

Реализация IPv4-адресации в сети начинается с настройки IPv4-адресов на хостах и маршрутизаторах, наряду с протоколами маршрутизации IPv4, изучающими маршруты к подсетям в этой сети. Этим темам и посвящена часть IV. Глава 15 сосредоточивается на маршрутизаторах Cisco, глава 16 — на IPv4-адресах маршрутизаторов и статической настройке маршрута IPv4. В главе 17 рассматривается настройка на маршрутизаторе протокола OSPF, позволяющего изучать маршруты. В главе 18 обсуждается реализация протокола IPv4 на хостах, а также некоторые простые инструментальные средства проверки его состояния.

Часть IV. Реализация IP-адресации версии 4

Глава 15

Работа с маршрутизаторами Cisco

В настоящей главе речь пойдет об установке маршрутизатора Cisco корпоративного класса с достаточной конфигурацией для работы. Напомним, что при покупке коммутатора LAN Cisco достаточно подключить к нему все кабели Ethernet, включить, и коммутатор заработает. Однако маршрутизаторы Cisco, используемые в компании, требуют предварительной настройки, прежде чем они смогут перенаправлять пакеты IPv4. В частности, следует указать, какие интерфейсы маршрутизатор должен использовать и какие IP-адреса на каждом из них должны быть.

В этой главе два главных раздела. В первом обсуждается физическая установка маршрутизатора Cisco корпоративного класса, а во втором рассматривается интерфейс командной строки (CLI) маршрутизатора Cisco, очень похожий на таковой у коммутатора Cisco. Сначала будут описаны сходства между CLI коммутатора и маршрутизатора, а затем настройка, необходимая для начала перенаправления пакетов IP на интерфейсах маршрутизатора.

В этой главе рассматриваются следующие экзаменационные темы
Работа сетей передачи данных IP
Назначение и функции различных сетевых устройств, таких как маршрутизаторы, коммутаторы, мосты и концентраторы.
Выбор компонентов сети, удовлетворяющих заданной спецификации.
Выбор подходящей среды, кабелей, портов и разъемов для подключения сетевых устройств Cisco к другим сетевым устройствам и хостам в сети LAN.
Технологии маршрутизации IP
Настройка и проверка применения CLI для установки базовой конфигурации маршрутизатора.
Команды Cisco IOS для базовой настройки маршрутизатора.
Защита сетевых устройств
Настройка и проверка средств защиты сетевых устройств.
Защита устройства паролем.
Привилегированный режим или защита.
Поиск и устранение неисправностей
Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Установка маршрутизаторов Cisco

Маршрутизаторы обеспечивают работу основной службы сетевого уровня (эталонной модели) — пересылку пакетов через сеть в сквозном режиме. Как упоминалось в главе 4, маршрутизаторы пересылают пакеты через разные физические сетевые соединения, например, через каналы Ethernet, последовательные каналы, среду Frame Relay, а также используют алгоритмы уровня 3 для принятия решения о том, куда именно следует отправить каждый пакет. В качестве напоминания отметим, что в главе 2 были описаны физические соединения с сетью Ethernet, а в главе 3 рассмотрены технологии WAN и их кабельные подключения.

В первом разделе подробно описан процесс установки маршрутизатора, сначала с точки зрения его развертывания в корпоративной сети, а потом — для подключения *малых и домашних офисов* (Small Office/Home Office — SOHO) к провайдеру Интернета с использованием высокоскоростных технологий абонентских каналов.

Установка маршрутизатора в корпоративной сети

В типичной сети крупного предприятия есть несколько централизованных *площадок* (site) и несколько небольших дистанционных филиалов. Для подключения устройств (компьютеров, IP-телефонов, принтеров и др.) на каждой площадке есть как минимум один коммутатор локальной сети. Кроме того, в сети каждой площадки должен быть как минимум один маршрутизатор, с помощью которого локальная сеть (LAN) будет подключена к распределенной сети (WAN). Канал WAN в такой структуре используется для подключения дистанционных площадок к центральному офису и для других телекоммуникационных нужд.

На рис. 15.1 и 15.2 приведены альтернативные способы представления частей корпоративной сети. Слева в обоих примерах представлен филиал с маршрутизатором и несколькими компьютерами конечных пользователей. У центральной площадки, показанной справа, в основном те же компоненты, плюс несколько серверов. Площадки соединены двухточечным последовательным каналом связи, проложенным между двумя маршрутизаторами. На первом рисунке отсутствует большинство подробностей, связанных с кабельным соединением (такой вид полезен при обсуждении общих концепций уровня 3), а на втором рисунке такие подробности представлены.

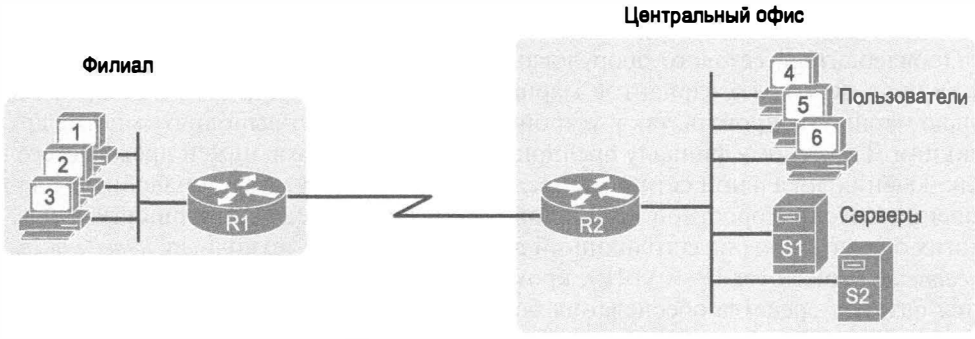


Рис. 15.1. Обобщенная схема корпоративной сети

Кабели Ethernet, показанные на рис. 15.2, вам должны быть уже знакомы. В частности, маршрутизаторы используют кабель Ethernet с той же схемой расположения выводов, что и компьютеры, поэтому каждый маршрутизатор использует кабель UTP с прямой схемой расположения выводов.

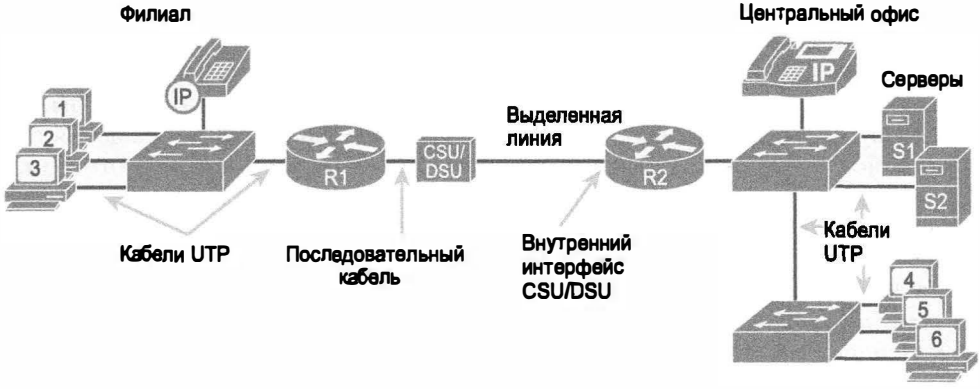


Рис. 15.2. Подробная схема корпоративной сети

Теперь рассмотрим аппаратные средства на концах последовательного канала связи, в частности, *модуль обслуживания канала/модуля обработки данных* (Channel Service Unit/Data Service Unit — CSU/DSU), располагающиеся на каждом конце последовательного канала связи. Они могут находиться вне маршрутизатора, как отдельное устройство (см. рис. 15.2, *слева*), или быть интегрированы в аппаратные средства последовательного интерфейса маршрутизатора (см. рис. 15.2, *справа*). У современных маршрутизаторов модуль CSU/DSU встроен в последовательный интерфейс.

И наконец, последовательному каналу связи требуется некий кабель. Проложенный телефонной компанией кабель WAN имеет обычно разъем RJ-48, совпадающий по размеру и форме с разъемом RJ-45. Кабель телефонной компании с разъемом RJ-48 подключают к устройству CSU/DSU. Кабель телефонной компании, показанный на рис. 15.2, в центральном офисе подключен непосредственно к последовательному интерфейсу маршрутизатора. В филиале кабель подключен к внешнему устройству CSU/DSU, которое подключено непосредственно к последовательному интерфейсу маршрутизатора другим последовательным кабелем. (Основы такого подключения см. в главе 3.)

Маршрутизаторы с интегрированными службами компании Cisco

Производители сетевого оборудования, в том числе и компания Cisco, обычно выпускают несколько вариантов маршрутизаторов, как устройств, которые могут только маршрутизировать, так и устройств, которые могут выполнять многие другие функции. Типичному филиалу предприятия маршрутизатор нужен прежде всего для подключения локальной сети к распределенной, а коммутатор понадобится для построения высокоскоростной локальной сети и подключения к маршрутизатору. Во многих организациях на сегодняшний день используется технология *передачи голоса по сети IP* (Voice over IP — VoIP), кроме того, практически любому офису понадобятся базовые средства обеспечения безопасности. (Одна из наиболее популярных служб, связанная с безопасностью, а именно технология *виртуальной частной сети*

(Virtual Private Network — VPN), описана в главе 5.) Вместо установки нескольких независимых устройств на одной площадке компания Cisco предлагает использовать устройства, которые могут работать одновременно и как маршрутизатор, и как коммутатор (рис. 15.2), а также обеспечивать дополнительные функции.

Развивая такую концепцию, компания Cisco производит несколько модельных серий маршрутизаторов, которые кроме маршрутизации могут выполнять также множество функций. У компании Cisco есть несколько серий маршрутизаторов, которые называют *маршрутизаторами с интегрированными службами* (Integrated Services Router — ISR), чтобы подчеркнуть тот факт, что множество служб интегрировано в одном устройстве. Тем не менее для упрощения материала и повышения эффективности обучения на курсах и экзаменах CCENT и CCNA маршрутизаторы и коммутаторы компании Cisco рассматриваются только как отдельные устройства, что облегчает понимание базовых технологий.

На рис. 15.3 показаны две фотографии маршрутизатора Cisco 2901 ISR, а также выделены некоторые важные компоненты.

Данная модель имеет два встроенных интерфейса GigabitEthernet и четыре модульных слота, позволяющих вставить дополнительные интерфейсные платы WAN (WAN Interface Card — WIC). Одна из плат WIC показана внизу рисунка, она устанавливается в один из этих четырех слотов. Маршрутизатор имеет и другие элементы, включая порты RJ-45, консольный порт и порт USB.

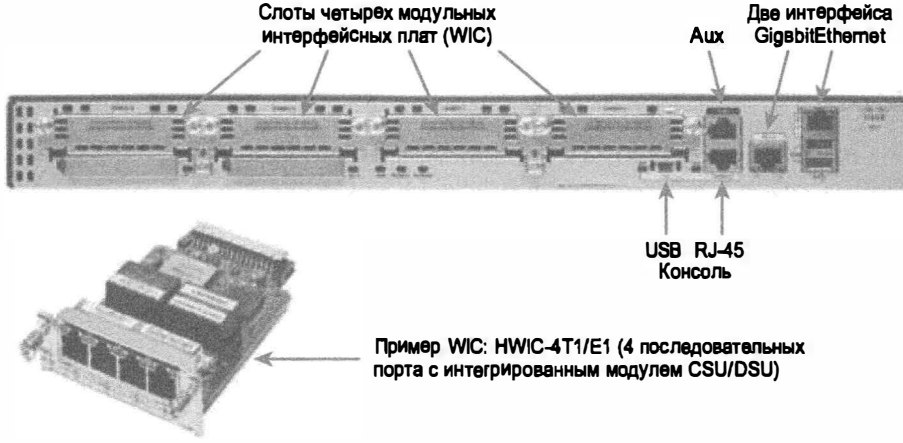


Рис. 15.3. Маршрутизатор Cisco модели 2901 ISR

Физическая установка устройства

Ознакомившись с кабелями на рис. 15.2 и подробностями аппаратных средств маршрутизатора на рис. 15.3, можно приступать к физической установке маршрутизатора. Для установки маршрутизатора выполните следующие действия.

Ключевая тема

Этапы установки маршрутизатора

Этап 1 Подключите кабели локальной сети к портам LAN устройства

Этап 2 Если используется внешний модуль CSU/DSU, подключите последовательный интерфейс маршрутизатора к модулю, а сам модуль — к линии от телекоммуникационной компании

Этап 3 Если используется встроенный модуль CSU/DSU, подключите последовательный интерфейс маршрутизатора к линии от телекоммуникационной компании

Этап 4 Подключите консольный порт маршрутизатора к персональному компьютеру (с помощью *обратного* (rollover) кабеля, т.е. консольного)

Этап 5 Подключите кабель питания к разъему питания устройства и настенной розетке

Этап 6 Включите питание маршрутизатора

Следует отметить, что указанные этапы установки практически полностью соответствуют этапам физической установки коммутатора, за исключением того, что у маршрутизаторов Cisco корпоративного уровня обычно есть выключатель, а у коммутатора нет.

Установка маршрутизатора доступа к сети

Маршрутизаторы играют ключевую роль в сетях SOHO, поскольку они соединяют подключенные к локальной сети устройства пользователей с высокоскоростной службой доступа к Интернету. Она может быть использована в малых и домашних сетях для доступа к центральной корпоративной сети, компании или учебного заведения.

Такие производители, как Cisco, выпускают сугубо маршрутизаторы и устройства, объединяющие функции маршрутизации с многими другими функциями. В целях обучения сначала рассмотрим каждую функцию в отдельности, а затем пример интегрированного сетевого устройства, объединяющего воедино множество функций.

Развертывание сети SOHO с отдельным коммутатором, маршрутизатором и кабельным модемом

На рис. 15.4 приведены схема сети, а также устройства и кабели, использующиеся для подключения сети SOHO к Интернету через высокоскоростную службу передачи данных *кабельного телевидения* (Cable TV — CATV). Следует отметить, что на схеме сети показан один вариант использования сетевых устройств и кабелей, хотя на самом деле их множество.

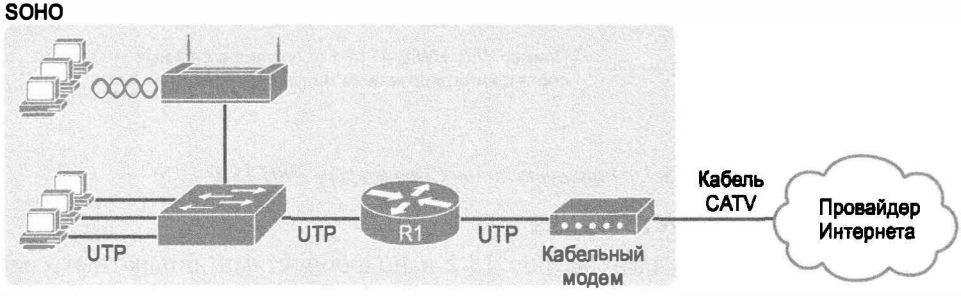


Рис. 15.4. Устройства сети SOHO и высокоскоростная служба передачи данных через канал кабельного телевидения

Схема сети на рис. 15.4 очень похожа на схему сети на рис. 15.2, где показан типичный проект сети и подключения офиса филиала крупного предприятия. Пользовательские рабочие станции подключены к коммутатору, а коммутатор подключен к интерфейсу Ethernet маршрутизатора. Другие устройства конечного пользователя используют беспроводную сеть LAN с беспроводной точкой доступа, также

подключенной к сети LAN Ethernet. Маршрутизатор осуществляет маршрутизацию, перенаправляя пакеты IP, как для проводных, так и беспроводных устройств.

Основное отличие между сетью SOHO, показанной на рис. 15.4, и инфраструктурой филиала предприятия (см. рис. 15.2) заключается в разных методах подключения к Интернету. Для канала Интернета, в котором используется инфраструктура кабельного телевидения или технология DSL, понадобится устройство, осуществляющее преобразование технологий и стандартов 1 и 2 уровней канала поставщика услуг в среду Ethernet, к которой подключается маршрутизатор. Такие устройства-преобразователи называют *кабельными модемами* (cable modem) и *модемами DSL* (DSL modem) соответственно.

Несмотря на то что принципы работы и детали кабельных модемов и модемов DSL существенно отличаются, они являются аналогами модулей CSU/DSU для последовательного интерфейса. Модуль CSU/DSU осуществляет преобразование сигнала и стандартов уровня 1 канала WAN оператора связи или телефонной компании в стандарт последовательного интерфейса маршрутизатора. Аналогично кабельный модем преобразует сигналы формата кабельного телевидения (т.е. стандарты уровня 1 и 2) в формат, приемлемый для маршрутизатора, например Ethernet. Модем DSL, соответственно, конвертирует сигналы формата DSL для абонентского телефонного канала в стандарт Ethernet.

Чтобы осуществить физическую установку сети SOHO и развернуть в ней устройства (см. рис. 15.4), понадобятся соответствующие кабели UTP для соединений Ethernet и кабельный выход оператора кабельного телевидения или телефонная линия (для канала DSL). Обратите внимание: маршрутизатору в таком случае достаточно всего двух интерфейсов Ethernet: один нужен для подключения коммутатора локальной сети, а другой для модема. Этапы установки сети SOHO перечислены ниже.

Этап 1 Соедините прямым (straight-through) кабелем UTP коммутатор и маршрутизатор

Этап 2 Соедините прямым кабелем UTP модем и маршрутизатор

Этап 3 Подключите консольный порт маршрутизатора к персональному компьютеру с помощью консольного кабеля

Этап 4 Подключите кабель питания к разъему питания устройства и настенной розетке

Этап 5 Включите питание маршрутизатора

Развертывание сети SOHO с интегрированными в одном устройстве коммутатором, маршрутизатором и кабельным модемом

В современных сетях SOHO обычно используются так называемые интегрированные устройства, т.е. устройства, объединяющие в себе несколько функций, а не отдельные коммутаторы, маршрутизаторы и т.п. (см. рис. 15.4). Фактически современные маршрутизаторы для сетей SOHO объединяют в себе сразу несколько устройств:

- маршрутизатор;
- коммутатор;
- кабельный модем или модем DSL;
- беспроводную точку доступа;
- аппаратный модуль шифрования трафика.

Современное высокоскоростное подключение к Интернету в сети SOHO, вероятно, скорее похоже на рис. 15.5, поскольку использует интегрированное устройством. Фактически, когда вы видите в магазине устройство потребительского уровня под названием "маршрутизатор", то это скорее всего одно из таких многофункциональных интегрированных устройств.

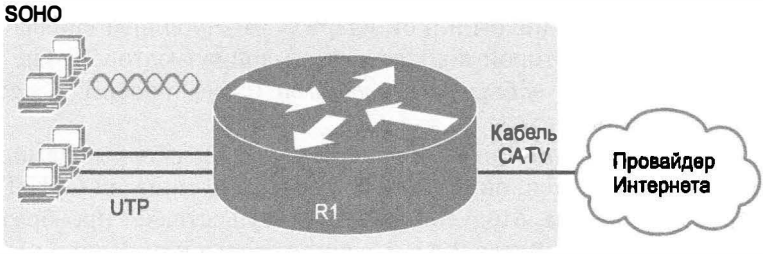


Рис. 15.5. Сеть SOHO с интегрированным устройством и подключением к оператору кабельного телевидения

Поддержка протокола IPv4 на маршрутизаторе Cisco

Благодаря большому количеству команд конфигурации и пользовательских команд маршрутизаторы поддерживают довольно много функций. Этот раздел знакомит с наиболее популярными командами, включая команды администрирования и команды, позволяющие применять протокол IPv4 на интерфейсах WAN и LAN.

ВНИМАНИЕ!

Документация маршрутизатора Cisco содержит справочник команд с индексом по каждой команде. Последняя версия операционной системы IOS насчитывает приблизительно 5000 команд CLI.

К счастью, изучить команды администрирования маршрутизатора довольно просто, поскольку многие маршрутизаторы и коммутаторы Cisco используют одинаковые команды. Например, команды консоли, vty и привилегированных паролей работают одинаково. Этот раздел начинается с обзора средств CLI, уже описанных в части II этой книги, и обзора тех из них, которые совпадают для маршрутизаторов и коммутаторов Cisco.

Остальная часть этого раздела посвящена базовым командам интерфейса маршрутизатора. В частности, рассматривается конфигурация, необходимая для запуска протокола IPv4 на интерфейсе маршрутизатора.

Сравнение интерфейса командной строки маршрутизатора и коммутатора

Ниже перечислены элементы конфигурации устройства, описанные в главе 8, которые в командной строке маршрутизатора и коммутатора выглядят абсолютно одинаково. Если читатель не очень хорошо помнит, как при помощи команд настраиваются разные функции, можно вернуться назад и просмотреть соответствующую главу еще раз.

Одинаково выглядят следующие команды настройки и компоненты интерфейса командной строки маршрутизаторов и коммутаторов.

Одинаковые команды и настройки интерфейса командной строки маршрутизатора и коммутатора

Ключевая тема

- команды обычного и привилегированного режимов ;
- команды переключения в режимы конфигурирования и команды выхода из него, в том числе команды `configure terminal`, `end`, `exit` и комбинация клавиш <Ctrl+Z>;
- настройка консольного, привилегированного паролей и аутентификации сеансов Telnet;
- настройка ключей шифрования протокола SSH, настройка имен и паролей пользователей;
- задание имени хоста (hostname) устройства и описаний для интерфейсов;
- настройки интерфейсов Ethernet, автоматическое определение параметров, а также команды `speed` и `duplex`;
- команды выключения (`shutdown`) и включения (`no shutdown`) интерфейсов;
- переключение из одних режимов конфигурирования в другие с помощью таких команд, как `line console 0` и `interface`;
- функция встроенной интерактивной подсказки командной строки, а также методы и клавиши повторного вызова команд;
- применение множества параметров команды `debug` для создания регистрационных сообщений об определенных событиях, чтобы любой пользователь мог контролировать эти сообщения, используя пользовательскую команду `terminal monitor`;
- режим начальной конфигурации, задающий пользователю набор вопросов и позволяющий создать простую начальную конфигурацию;
- функции различных конфигурационных файлов: `startup-config` в памяти NVRAM, `running-config` в оперативной памяти (RAM); взаимодействие с внешними хранилищами (например, с сервером TFTP); а также принципы использования команды `copy` для перемещения конфигурационных файлов и образов операционной системы Cisco IOS;

На первый взгляд может показаться, что все важные команды уже были описаны в главе 8, и это действительно так — многие команды начального конфигурирования устройства там были рассмотрены подробно. Тем не менее некоторые службы и функции работают в коммутаторах и маршрутизаторах совсем по-разному, а именно:

Команды и настройки (см. главу 8), которые отличаются в маршрутизаторах и коммутаторах

Ключевая тема

- несколько отличается настройка IP-адресов;
- в маршрутизаторах есть специальный дополнительный порт (Auxiliary — AUX), предназначенный для подключения внешнего модема и телефонной линии, чтобы была возможность получить дистанционный доступ к командной строке устройства через телефонную сеть.

Кроме двух указанных выше моментов, коммутаторы и маршрутизаторы отличаются по своим функциям, следовательно, отличаются команды в интерфейсе командной строки. Например, коммутаторы Cisco уровня 2 поддерживают команду `show mac address-table`, но не поддерживают команду `show ip route`, выводящую маршруты IP. Некоторые маршрутизаторы Cisco способны осуществлять маршрутизацию IP, но не коммутаторы уровня 2, поэтому они поддерживают команду `show ip route`, но не команду `show mac address-table`.

Интерфейсы маршрутизатора

Между коммутаторами и маршрутизаторами Cisco есть одно незначительное различие — маршрутизаторы поддерживают более широкое разнообразие интерфейсов. Современные коммутаторы LAN поддерживают интерфейсы Ethernet LAN различных скоростей. Маршрутизаторы поддерживают множество интерфейсов других типов, включая последовательные интерфейсы, интерфейсы кабельного телевидения, DSL и другие, не упомянутые в этой книге.

У большинства маршрутизаторов Cisco есть по крайней мере один интерфейс Ethernet. Большинство этих интерфейсов Ethernet поддерживает несколько скоростей и использует автопереговоры, таким образом, для совместимости операционная система маршрутизатора именует эти интерфейсы на основании их максимальной скорости. Например, интерфейсы Ethernet на 10 Мбит/с настраиваются только командой конфигурации `interface ethernet` *номер*, интерфейсы 10/100 — командой `interface fastethernet` *номер*, а интерфейсы 10/100/1000 — командой `interface gigabitethernet` *номер*.

Последовательные интерфейсы являются вторым по распространенности типом интерфейса маршрутизатора. Как уже упоминалось в главе 3, маршрутизаторы Cisco используют последовательные интерфейсы для подключения к последовательному каналу связи. Каждый двухточечный последовательный канал связи может использовать *высокоуровневый протокол управления каналом* (High-Level Data Link Control — HDLC) или *протокол двухточечного соединения* (Point-to-Point Protocol — PPP).

Во многих командах маршрутизаторов интерфейсы именуются сначала по типу (Ethernet, Fast Ethernet, Serial и т.д.), а затем по индивидуальному номеру. Номера интерфейсов маршрутизаторов могут быть одним числом, двумя или тремя числами, разделенными косой чертой. Например, все три приведенных ниже команды конфигурации правильны (по крайней мере, для одной из моделей маршрутизатора Cisco).

```
interface ethernet 0
interface fastEthernet 0/1
interface gigabitethernet 0/0
interface serial 1/0/1
```

Двумя наиболее распространенными командами отображения состояния интерфейсов являются `show ip interface brief` и `show interfaces`. Первая из них отображает список интерфейсов (по строке на каждый) с дополнительной информацией, включая его IP-адрес и состояние. Вторая перечисляет интерфейсы, но уже с большим объемом информации. Пример 15.1 демонстрирует пример каждой из этих команд.

Пример 15.1. Получение информации об интерфейсах маршрутизатора

```
R1# show ip interface brief
Interface                  IP-Address OK? Method Status     Protocol
Embedded-Service-Engine0/0 unassigned YES NVRAM  adm. down  down
GigabitEthernet0/0         172.16.1.1 YES NVRAM  down       down
GigabitEthernet0/1         unassigned YES manual adm. down  down
Serial0/0/0                172.16.4.1 YES NVRAM  up         up
Serial0/0/1                172.16.5.1 YES NVRAM  up         up
Serial0/1/0                unassigned YES NVRAM  up         up
Serial0/1/1                unassigned YES NVRAM adm. down   down

R1# show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Description: Link in lab to R2's S0/0/1
  Internet address is 172.16.4.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:03, output 00:00:06, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    42 packets input, 3584 bytes, 0 no buffer
    Received 42 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    41 packets output, 3481 bytes, 0 underruns
    0 output errors, 0 collisions, 4 interface resets
    3 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
    DCD=up DSR=up DTR=up RTS=up CTS=up
```

ВНИМАНИЕ!

Команды для работы с интерфейсами можно значительно сократить. Например, запись `sh int fa0/0` является полным аналогом команды `show interfaces fastethernet 0/0`. Зачастую, заглядывая через чье-то плечо, инженеры говорят что-либо похожее на "Покажи-ка мне `show int F-A` и т.д.", а не пытаются произнести полный вариант команды.

Обратите также внимание на то, что команда `show interfaces` выводит текстовое описание интерфейса примерно в 3-й строке, если он настроен. В данном случае интерфейс S0/0/0 был предварительно настроен командой `description Link in lab to R2's S0/0/1` в режиме конфигурации интерфейса. Подкоманда интерфейса `description` позволяет добавить небольшое примечание о подключенном к интерфейсу маршрутизатора устройстве, а команда `show interfaces` отображает эту информацию.

Коды состояний интерфейсов

У каждого интерфейса есть два *кода состояния интерфейса* (interface status code). Чтобы интерфейс был пригодным для использования, оба кода должны быть в состоянии "up". Первый код состояния относится к уровню 1, а второй обычно (но не всегда) указывает состояние протокола канального уровня. В табл. 15.1 описаны два кода состояний и их значение.

Ключевая тема **Таблица 15.1. Коды состояния интерфейсов и их значение**

Название	Местоположение	Значение
Состояние линии	Первый код состояния	Описывает состояние уровня 1, например, подключен ли кабель, правильный ли это кабель, включено ли питание устройства на другом конце канала
Состояние протокола	Второй код состояния	Описывает состояние уровня 2. В этом поле всегда отображается слово "down", если не работает первый уровень (его состояние — "down"). Если линия находится в состоянии "up", состояние протокола "down" обычно означает ошибки в настройке протокола канального уровня

В табл. 15.2 приведено несколько комбинаций кодов состояния интерфейса, в порядке от полностью отключенного до полностью рабочего состояния.

Ключевая тема **Таблица 15.2. Типичные комбинации кодов состояния интерфейсов**

Состояние линии	Состояние протокола	Типичные причины
administratively down	down	В конфигурации интерфейса введена команда shutdown
down, down	down	Интерфейс не отключен, но есть проблемы на физическом уровне. Например, в интерфейс не включен кабель; если используется канал Ethernet, то интерфейс коммутатора на другом конце канала выключен или выключено питание самого коммутатора
Up, down	down	Практически всегда эти коды состояний указывают на проблемы канального уровня, зачастую — на проблемы в конфигурации. Например, для последовательных интерфейсов на одном конце канала может быть указана инкапсуляция PPP, а на другом — HDLC
Up, up	up	Работают уровни 1 и 2 данного интерфейса

Пример 15.1 демонстрирует несколько команд show ip interface brief для трех интерфейсов, приведенных ниже. У каждого интерфейса разные комбинации кодов состояния, а также их наиболее вероятные причины.

Интерфейс G0/0 находится в состоянии down/down, поскольку в данном случае к нему не был подключен кабель.

Интерфейс G0/1 находится в состоянии administratively down/down, поскольку в его конфигурацию включена команда shutdown.

Интерфейс S0/0/0 находится в состоянии up/up, поскольку он подключен последовательным кабелем к другому работающему маршрутизатору.

IP-адрес интерфейса маршрутизатора

Прежде чем маршрутизаторы Cisco корпоративного уровня смогут выполнять свою основную задачу — маршрутизацию пакетов IP, — необходимо изменить их стандартную конфигурацию. Для подготовки маршрутизатора к перенаправлению пакетов IPv4 на интерфейсе необходимо включить интерфейс и присвоить ему IPv4-адрес в связи со следующими фактами.

- Большинство интерфейсов маршрутизаторов Cisco изначально отключено (shutdown), их необходимо включить подкомандой интерфейса no shutdown.
- Маршрутизаторы Cisco не направляют пакеты IP ни на интерфейс, ни через него, пока для него не заданы IP-адрес и маска (изначально они не заданы).
- Маршрутизаторы Cisco пытаются перенаправлять пакеты IP на любых интерфейсах, находящихся в состоянии up/up, имеют IP-адреса и маски подсети. (Маршрутизаторы изначально позволяют маршрутизацию IPv4, поскольку стандартно введена глобальная команда конфигурации ip routing.)

Для настройки адреса и маски достаточно использовать подкоманду интерфейса ip address *адрес маска*. На рис. 15.6 приведена простая сеть IPv4, используемая в нескольких примерах создания подсетей в части III этой книги. Рисунок демонстрирует IPv4-адреса на маршрутизаторе R1, а соответствующая настройка приведена в примере 15.2.

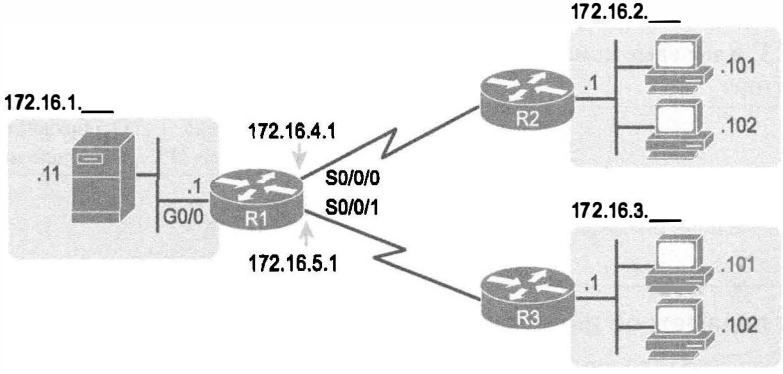


Рис. 15.6. IPv4-адреса, используемые в примере 15.2

Пример 15.2. Настройка IP-адресов на маршрутизаторах Cisco

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1config)# interface G0/0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface S0/0/0
```

```
R1(config-if)# ip address 172.16.4.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface S0/0/1
R1(config-if)# ip address 172.16.5.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# ^Z
R1#
```

Пример 15.3 демонстрирует вывод команды `show protocols` и подтверждает состояние каждого из трех интерфейсов маршрутизатора R1, приведенных на рис. 15.6, их IP-адреса и маски.

Пример 15.3. Проверка IP-адресов на маршрутизаторе Cisco

```
R1# show protocols
Global values:
  Internet Protocol routing is enabled
Embedded-Service-Engine0/0 is administratively down, line protocol is down
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 172.16.1.1/24
GigabitEthernet0/1 is administratively down, line protocol is down
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.4.1/24
Serial0/0/1 is up, line protocol is up
  Internet address is 172.16.5.1/24
Serial0/1/0 is administratively down, line protocol is down
Serial0/1/1 is administratively down, line protocol is down
```

При проверке работоспособности маршрутизатора в первую очередь выясняют состояние интерфейса, а также правильность IP-адреса и маски. В примерах 15.1 и 15.3 представлены ключевые команды `show`, а в табл. 15.3 кратко резюмируются эти команды и типы отображаемой ими информации.

Таблица 15.3. Ключевые команды, отображающие состояние интерфейсов маршрутизатора

Команда	Строк вывода на интерфейс	Отображаемая конфигурация IP	Отображается ли состояние интерфейса?
`Show ip interface brief`	1	Адрес	Да
`show protocols` [*тип номер*]	1 или 2	Адрес/маска	Да
`show interfaces` [*тип номер*]	Много	Адрес/маска	Да

Установка параметров `Bandwidth` и `Clock Rate` для интерфейсов

Компания Cisco уделяет не много внимания подробностям технологий WAN на экзаменах CCENT, в отличие от экзаменов ICND2 и CCNA R/S. Но если решено создать собственную лабораторную сеть из реальных устройств, то необходимо знать больше о последовательных каналах связи. Им посвящен последний раздел главы.

Как упоминалось в главе 3, разнообразие технологий и скоростей каналов WAN очень велико. Чтобы использовать разные скорости передачи данных, маршрутизаторы работают в качестве ведомого устройства и получают настройки скорости от модуля CSU/DSU в ходе *синхронизации* (clocking). В результате последовательные каналы маршрутизатора работают без дополнительной настройки, автоматического

определения скорости канала и т.п. Устройство CSU/DSU всегда "знает" скорость работы телекоммуникационного канала, пересылает синхроимпульсы по кабелю маршрутизатору, а последний подстраивает свой интерфейс согласно таким импульсам. Фактически модуль CSU/DSU указывает маршрутизатору, когда следует отправить следующий бит по кабелю и принять бит, а маршрутизатор просто слепо следует таким инструкциям.

Чтобы создать последовательный канал связи в домашней лаборатории, можно использовать платы последовательного интерфейса маршрутизаторов, которые обычно используют внешний модуль CSU/DSU. Так можно получить последовательный канал связи и без двух модулей CSU/DSU. Эта концепция приведена на рис. 3.5 в главе 3 и повторена здесь на рис. 15.7. Чтобы канал связи заработал, используют два последовательных кабеля: кабель DTE и кабель DCE, которые перекрещивают передающую и принимающую пары на кабелях.

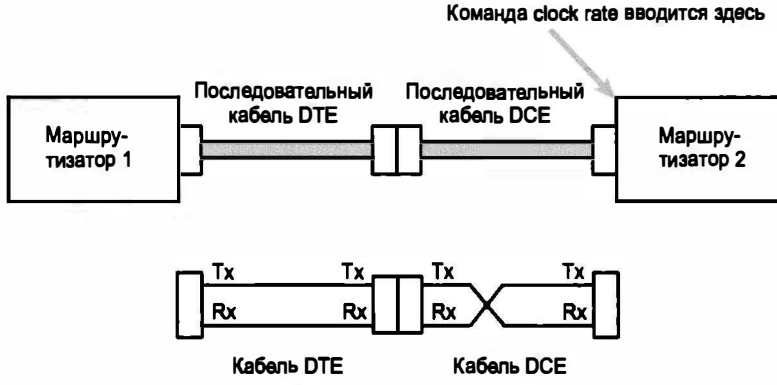


Рис. 15.7. Лабораторный последовательный канал связи

После установки кабелей для правильной работы соединения добавьте подкоманду интерфейса `clock rate`. Эта команда задает маршрутизатору скорость передачи битов на последовательном канале связи, как показано на рис. 15.7. Команда `clock rate` необязательна в реальных последовательных каналах связи, поскольку модуль CSU/DSU обеспечивает синхронизацию сам. Но без реального блока CSU/DSU на канале связи маршрутизатор с кабелем DCE должен использовать функцию синхронизации, а команда `clock rate` включает ее на маршрутизаторе.

ВНИМАНИЕ!

Некоторые версии операционной системы IOS автоматически реализуют команду `clock rate 20000000` на последовательных интерфейсах с кабелем DCE. Это весьма удобно, хотя такая скорость может оказаться слишком высокой для некоторых типов последовательных кабелей, поэтому учитывайте данный факт в лабораторной модели.

Пример 15.4 демонстрирует применение команды `clock rate` для того же маршрутизатора R1, что и в примере 15.2. В конце примера этот маршрутизатор проверяется на возможность использования команды `clock rate` при помощи команды `show controllers`. Вывод этой команды подтверждает, что маршрутизатор R1 имеет кабельное соединение V.35 DCE.

ВНИМАНИЕ!

В примере 15.4 опущена большая часть вывода команды `show running-config`, т.е. убрано все, что не относится к обсуждаемой теме.

Пример 15.4. Настройка маршрутизатора R1 с использованием команды `clock rate`

```
R1# show running-config
! Остальные строки опущены для краткости
interface Serial0/0/0
  ip address 172.16.4.1 255.255.255.0
  clock rate 20000000
!
interface Serial0/0/1
  ip address 172.16.5.1 255.255.255.0
  clock rate 128000

! Остальные строки опущены для краткости

R1# show controllers serial 0/0/1
Interface Serial0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 128000
idb at 0x8169BB20, driver data structure at 0x816A35E4
! Остальные строки опущены для краткости
```

ВНИМАНИЕ!

Команда `clock rate` не позволяет задать любую скорость, так как список скоростей зависит от конкретного маршрутизатора.

Вспомогательный порт маршрутизатора (Aux)

И у маршрутизаторов, и у коммутаторов есть консольный порт, обеспечивающий административный доступ, а у маршрутизаторов есть дополнительный физический порт: вспомогательный порт (Aux). Порт Aux обычно используется для телефонного вызова, позволяющего подключиться к маршрутизатору и ввести команды CLI.

Порт Aux работает как канал консоли, но подключение осуществляется через кабель с внешним аналоговым модемом, который в свою очередь подключен к телефонной линии. Для дистанционной связи с маршрутизатором инженер использует компьютер, эмулятор терминала и модем. Установив соединение, инженер может использовать эмулятор терминала для доступа к интерфейсу CLI маршрутизатора, вначале в пользовательском режиме, как обычно.

Параметры порта Aux задаются после перехода в режим конфигурирования соответствующей линии с помощью команды `line aux 0`. В режиме конфигурирования линии порта Aux можно ввести многие команды, которые обсуждались в главе 8. Например, можно настроить простую аутентификацию с помощью команд `login` и `password`; в результате при входящем дистанционном соединении у пользователя будут запрашиваться имя и пароль.

Определение состояния при помощи команды `show version`

И наконец, еще одна команда, `show version`, выводит большое количество важных сведений о маршрутизаторе. Она выводит текущую версию IOS маршрутизатора и многие другие очень важные подробности: как долго маршрутизатор работает, почему операционная система IOS была перезагружена в последний раз, какой файл использовался для загрузки IOS и какие интерфейсы установлены на маршрутизаторе. Она выводит также подробности об объемах NVRAM, RAM и флеш-памяти.

В примере 15.5 выделены строки с наиболее интересными подробностями.

Пример 15.5. Отображение версии IOS и других подробностей командой `show version`

```
R1# show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version
15.2(4)M1, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 26-Jul-12 20:54 by prod_rel_team

ROM: System Bootstrap, Version 15.0(1r)M15, RELEASE SOFTWARE (fc1)

R1 uptime is 3 days, 4 hours, 19 minutes
System returned to ROM by reload at 14:00:46 UTC Sat Oct 13 2012
System image file is "flash:c2900-universalk9-mz.SPA.152-4.M1.bin"
Last reload type: Normal Reload
Last reload reason: Reload Command

! Остальные строки опущены для краткости

If you require further assistance please contact us by sending email to
export@cisco.com.

Cisco CISCO2901/K9 (revision 1.0) with 483328K/40960K bytes of memory.
Processor board ID FTX1628838P
2 Gigabit Ethernet interfaces
4 Serial(sync/async) interfaces
1 terminal line
DRAM configuration is 64 bits wide with parity enabled.
255K bytes of non-volatile configuration memory.
3425968K bytes of USB Flash usbflash1 (Read/Write)
250880K bytes of ATA System CompactFlash 0 (Read/Write)

License Info:

! Остальные строки опущены для краткости

Configuration register is 0x2102
```

Обзор

Резюме

- Маршрутизаторы обеспечивают работу основной службы сетевого уровня (эталонной модели) — пересылку пакетов через сеть в сквозном режиме.
- В типичной сети крупного предприятия есть несколько централизованных площадок (site) и несколько небольших дистанционных филиалов. Для подключения устройств (компьютеров, IP-телефонов, принтеров и др.) на каждой площадке есть как минимум один коммутатор локальной сети. Кроме того, в сети каждой площадки должен присутствовать как минимум один маршрутизатор, с помощью которого локальная сеть (LAN) будет подключена к распределенной сети (WAN). Канал WAN в такой структуре используется для подключения дистанционных площадок к центральному офису и для других телекоммуникационных нужд.
- Типичный филиал корпорации нуждается в маршрутизаторе для соединения сетей WAN/LAN и коммутатора LAN, поддерживающего высокопроизводительную локальную сеть и подключение к маршрутизатору и WAN.
- Для установки маршрутизатора выполните следующие действия.

  Этап 1 Подключите кабели локальной сети к портам LAN устройства

  Этап 2 Если используется внешний модуль CSU/DSU, подключите последовательный интерфейс маршрутизатора к модулю, а сам модуль — к линии от телекоммуникационной компании

  Этап 3 Если используется встроенный модуль CSU/DSU, подключите последовательный интерфейс маршрутизатора к линии от телекоммуникационной компании

  Этап 4 Подключите консольный порт маршрутизатора к персональному компьютеру (с помощью *обратного* (rollover) кабеля, т.е. консольного)

  Этап 5 Подключите кабель питания к разъему питания устройства и настенной розетке

  Этап 6 Включите питание маршрутизатора
- Современные маршрутизаторы для сетей SOHO объединяют в себе сразу несколько устройств:
  - маршрутизатор;
  - коммутатор;
  - кабельный модем или модем DSL;
  - беспроводную точку доступа;
  - аппаратный модуль шифрования трафика.
- Одинаково выглядят следующие команды настройки и компоненты интерфейса командной строки маршрутизаторов и коммутаторов:
  - команды обычного и привилегированного режимов;
  - команды переключения в режимы конфигурирования и команды выхода из него, в том числе команды `configure terminal`, `end`, `exit` и комбинация клавиш <Ctrl+Z>;
  - настройка консольного, привилегированного паролей и аутентификации сеансов Telnet;

- настройка ключей шифрования протокола SSH, настройка имен и паролей пользователей;
- задание имени хоста устройства и описаний для интерфейсов;
- настройки интерфейсов Ethernet, автоматическое определение параметров, а также команды `speed` и `duplex`;
- команды выключения (`shutdown`) и включения (`no shutdown`) интерфейсов;
- переключение из одних режимов конфигурирования в другие с помощью таких команд, как `console 0` и `interface`;
- функция встроенной интерактивной подсказки командной строки, а также методы и клавиши повторного вызова команд;
- применение множества параметров команды `debug` для создания регистрационных сообщений об определенных событиях, чтобы любой пользователь мог контролировать эти сообщения, используя пользовательскую команду `terminal monitor`;
- режим начальной конфигурации, задающий пользователю набор вопросов и позволяющий создать простую начальную конфигурацию;
- функции различных конфигурационных файлов: `startup-config` в памяти NVRAM, `running-config` в оперативной памяти (RAM); взаимодействие с внешними хранилищами (например, с сервером TFTP); а также принципы использования команды `copy` для перемещения конфигурационных файлов и образов операционной системы Cisco IOS;

- Между коммутаторами и маршрутизаторами Cisco есть одно незначительное различие — маршрутизаторы поддерживают более широкое разнообразие интерфейсов. Современные коммутаторы LAN поддерживают интерфейсы Ethernet LAN различных скоростей. Маршрутизаторы поддерживают множество интерфейсов других типов, включая последовательные интерфейсы, интерфейсы кабельного телевидения, DSL и другие, не упомянутые в этой книге.
- У каждого интерфейса есть два кода состояния интерфейса. Чтобы интерфейс был пригодным для использования, оба кода должны быть в состоянии "`up`".
  - Первый код состояния — состояние линии — описывает состояние уровня 1, например, подключен ли кабель, правильный ли это кабель, включено ли питание устройства на другом конце канала?
  - Второй код состояния — состояние протокола — описывает состояние уровня 2. В этом поле всегда отображается слово "`down`", если не работает первый уровень (его состояние — "`down`"). Если линия находится в состоянии "`up`", состояние протокола "`down`" обычно означает ошибки в настройке протокола канального уровня.
- Для подготовки маршрутизатора к перенаправлению пакетов IPv4 на интерфейсе необходимо включить интерфейс и присвоить ему IPv4-адрес в связи со следующими фактами.
  - Большинство интерфейсов маршрутизаторов Cisco изначально отключено (`shutdown`), их необходимо включить подкомандой интерфейса `no shutdown`.

- Маршрутизаторы Cisco не направляют пакеты IP ни на интерфейс, ни через него, пока для него не заданы IP-адрес и маска (изначально они не заданы).
- Маршрутизаторы Cisco пытаются перенаправлять пакеты IP на любых интерфейсах, находящихся в состоянии `up/up` и обладающих IP-адресами и масками подсети. (Маршрутизаторы изначально позволяют маршрутизацию IPv4, поскольку стандартно введена глобальная команда конфигурации `ip routing`.)

- И у маршрутизаторов, и у коммутаторов есть консольный порт, обеспечивающий административный доступ, а у маршрутизаторов есть дополнительный физический порт — вспомогательный порт (Aux). Порт Aux обычно используется для телефонного вызова, позволяющего подключиться к маршрутизатору и ввести команды CLI.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Какие этапы при установке устройства присущи маршрутизаторам компании Cisco, но отсутствуют у коммутаторов? (Выберите два ответа.)

   А) Подключение кабелей Ethernet.

   Б) Подключение последовательных (serial) кабелей.

   В) Подключение к консольному порту.

   Г) Подключение питания.

   Д) Переключение выключателя устройства в положение "включен".

2. Какую из указанных ниже команд можно встретить в интерфейсе командной строки маршрутизатора, но не коммутатора?

   А) Команда `clock rate`.

   Б) Команда `ip address` *маска адрес*.

   В) Команда `ip address dhcp`.

   Г) Команда `interface vlan 1`.

3. Ваша компания только что приобрела два маршрутизатора компании Cisco для использования на лабораторном стенде. Устройства подключены к разным сегментам локальной сети интерфейсами Fa0/0, а последовательные интерфейсы (serial) подключены друг к другу напрямую. Какие из указанных ниже действий не нужно выполнять, чтобы обеспечить передачу пакетов IPv4? (Выберите два ответа.)

   А) Настроить IP-адреса на интерфейсах FastEthernet и последовательных интерфейсах обоих маршрутизаторов.

   Б) Настроить команду `bandwidth` на последовательном интерфейсе для одного маршрутизатора.

   В) Настроить команду `clock rate` на последовательном интерфейсе одного маршрутизатора.

   Г) Задать с помощью команды `description` описание на нужных интерфейсах FastEthernet и последовательных интерфейсах обоих маршрутизаторов.

4. Вывод команды `show ip interface brief` маршрутизатора показывает коды состояний "выключено" (`"down"` и `"down"`) для интерфейса Serial 0/0. В чем может заключаться причина?

   А) В данном интерфейсе введена команда `shutdown`.

   Б) В интерфейсе маршрутизатора было задано использование технологии Frame Relay, но на другом конце канала используется инкапсуляция PPP.

   В) В соответствующий последовательный интерфейс не включен последовательный кабель.

   Г) Оба маршрутизатора на концах канала подключены к работающей среде передачи данных (через модуль CSU/DSU), но только на одном из них установлен IP-адрес.

5. Какая из указанных ниже команд не покажет настройки IP-адресов и масок в устройстве? (Выберите два ответа.)

   А) `show running-config`.

   Б) `show protocols` *тип номер*.

   В) `show ip interface brief`.

   Г) `show interfaces`.

   Д) `show version`.

6. Чем отличаются интерфейсы командной строки маршрутизатора и коммутатора компании Cisco?

   А) Командами для конфигурирования простой процедуры проверки паролей для консоли.

   Б) Количеством заданных IP-адресов.

   В) Заданием имени хоста (hostname).

   Г) Заданием описаний для интерфейсов.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. ниже.

Таблица 15.4. Ключевые темы главы 15

Элемент	Описание	Страница
Список	Этапы установки маршрутизатора	447
Список	Одинаковые команды и настройки интерфейса командной строки маршрутизатора и коммутатора	451
Список	Команды и настройки (см. главу 8), которые отличаются в маршрутизаторах и коммутаторах	451
Табл. 15.1	Коды состояния интерфейсов и их значение	454
Табл. 15.2	Типичные комбинации кодов состояния интерфейсов	454

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

ширина полосы пропускания (bandwidth), тактовая частота (clock rate), образ IOS (IOS image)

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 15.5 приведен список команд конфигурации, а в табл. 15.6 пользовательские команды главы. Фактически команды стоит запомнить, чтобы лучше понять материал главы и выполнить задачи по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 15.5. Команды конфигурации главы 15

Команда	Описание
`interface` *тип номер*	Глобальная команда, переводящая пользователя в режим конфигурации заданного интерфейса
`ip address` *адрес маска_подсети*	Подкоманда интерфейса, устанавливающая IPv4-адреса и маску маршрутизатора
`[no] shutdown`	Подкоманда интерфейса, включающая (`no shutdown`) или отключающая (`shutdown`) интерфейс
`duplex{ full | half | auto }`	Подкоманда, устанавливающая дуплексный режим работы интерфейса; если указан параметр `auto`, выполняется автосогласование дуплексности
`speed { 10 | 100 | 1000 }`	Подкоманда на (10/100/1000) Гигабит маршрутизатора, задающая скорость передачи и получения данных
`clock rate` *значение*	Задает частоту синхроимпульсов и скорость передачи данных через интерфейс. Эта команда применима только для интерфейсов, к которым подключен кабель DCE. Команда вводится в режиме конфигурации интерфейса
`description` *текст*	Подкоманда интерфейса для ввода строки текста, документирующего информацию о специфическом интерфейсе

Таблица 15.6. Пользовательские команды главы 15

Команда	Описание
`show interfaces [тип номер]`	Отображает подробную информацию о состоянии интерфейса, настройках и его счетчиках
`show ip interface brief`	Отображает строку информации о каждом интерфейсе, в том числе IP-адрес, код состояния линии и протокола, а также метод получения IP-адреса (вручную или через протокол DHCP)
`show protocols тип номер`	Отображает строку информации о каждом интерфейсе, в том числе IP-адрес, маску и коды состояний линии и протокола
`show controllers [тип номер]`	Выводит множество информации об интерфейсе, а именно об аппаратном контроллере. Для последовательных интерфейсов позволяет определить, подключен кабель DTE или DCE
`show version`	Выводит версию операционной системы IOS, выполняющейся на маршрутизаторе в настоящий момент, а также множество других фактов об аппаратных средствах и программном обеспечении, установленных в настоящее время на маршрутизаторе

Ответы на контрольные вопросы:

1 Б и Д. **2** А. **3** Б и Г. **4** В. **5** В и Д. **6** Б.

ГЛАВА 16

Настройка IPv4-адресов и маршрутов

Маршрутизаторы перенаправляют пакеты IPv4. Это простое утверждение имеет большой скрытый смысл. Для перенаправления пакетов маршрутизаторы осуществляют процесс маршрутизации, который полагается на информацию о маршрутах IP. Каждый маршрут IP задает назначение: сеть IP, подсеть IP или некую другую группу IP-адресов. Каждый маршрут имеет также инструкции, указывающие маршрутизатору направление перенаправления пакетов, посланных на адрес в этой сети или подсети IP. Чтобы маршрутизатор смог выполнять задачу маршрутизации пакетов, он должен иметь подробный и точный список маршрутов IP.

Для добавления маршрутов IPv4 в таблицы маршрутизации маршрутизаторы используют три метода. Сначала они изучают *подключенные маршруты* (connected route), т.е. маршруты для подсетей, подключенных к интерфейсу маршрутизатора. Маршрутизаторы могут также использовать *статические маршруты* (static route), создаваемые при помощи команды конфигурации `ip route`, непосредственно помещающей маршрут в таблицу маршрутизации IPv4. Кроме того, маршрутизаторы могут использовать протокол маршрутизации, по которому они оповещают друг друга обо всех известных маршрутах, чтобы все маршрутизаторы могли изучить все маршруты ко всем сетям и подсетям.

Глава начинается с продолжения знакомства с процессом маршрутизации IP, полагающимся на маршруты IP. Это продолжение обсуждения, начатого в главе 4, а также более глубокое обсуждение взаимосвязанных концепций, включая информацию об одиночном маршруте IP. Затем речь пойдет о подключенных маршрутах, включая варианты подключенных маршрутов к сетям VLAN, к магистральным каналам маршрутизаторов VLAN и подключенных маршрутов на коммутаторах уровня 3.

В заключительном разделе рассматриваются статические маршруты, позволяющие инженеру самостоятельно добавлять маршрут (маршруты) к таблице маршрутизации IP маршрутизатора. В разделе, посвященном статическим маршрутам, демонстрируется также настройка статического стандартного маршрута, используемого при отсутствии подходящего маршрута для пакета IP. Динамическая маршрутизация с использованием открытого протокола поиска первого кратчайшего маршрута (OSPF) рассматривается в главе 17.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

- Передача данных между двумя хостами по сети.
- Технологии маршрутизации IP
- Базовые концепции маршрутизации:
  - CEF.
  - Передача пакета.
  - Процесс поиска маршрутизатора.
- Настройка и проверка применения CLI для установки базовой конфигурации маршрутизатора.
  - Команды Cisco IOS для базовой настройки маршрутизатора.
- Настройка и проверка состояния интерфейса Ethernet.
- Проверка конфигурации маршрутизатора и сетевого подключения.
  - Команды Cisco IOS для просмотра базовой информации маршрутизатора и сетевого подключения.
- Настройка и проверка конфигурации маршрутизации для статического или стандартного маршрута согласно заданным требованиям маршрутизации.
- Различия методов маршрутизации и протоколов маршрутизации:
  - Ближайшая точка перехода.
  - Таблица IP-маршрутизации.
- Настройка и проверка маршрутизации между VLAN (router on a stick).
  - Субинтерфейсы.
  - Восходящая маршрутизация.
  - Инкапсуляция.
- Настройка интерфейсов SVI.

Основные темы

Маршрутизация IP

Маршрутизация IP (процесс перенаправления пакетов IP) обеспечивает доставку пакетов через все сети TCP/IP с устройства, создавшего пакет IP, на устройство его получателя. Другими словами, маршрутизация IP доставляет пакеты IP с хоста отправителя на хост получателя.

Полный процесс сквозной маршрутизации использует логику сетевого уровня на хостах и маршрутизаторах. Для создания и перенаправления пакета IP на стандартный шлюз хоста (стандартный маршрутизатор) передающий хост использует концепции уровня 3. Когда, принимая решение о перенаправлении пакета IP, маршрутизатор сравнивает адрес получателя в пакете с таковым в таблице маршрутизации, также используется логика уровня 3.

Процесс маршрутизации полагается также на физические свойства каждого канала связи. Маршрутизация IP использует последовательные каналы связи, локальные сети Ethernet, беспроводные локальные сети и много других сетей, реализующих стандарты физического уровня и канал связи. Эти низкоуровневые устройства и протоколы перемещают пакеты IP по сети TCP/IP, инкапсулируя и передавая пакеты во фреймах канального уровня.

Итак, резюмировав ключевые концепции маршрутизации IP, представленные в главе 4, перейдем к обсуждению следующего этапа на основании знаний, полученных в частях II и III.

ВНИМАНИЕ!

Иногда ошибочно утверждается, что термин "маршрутизация IP" (IP routing) подразумевает также функцию динамического изучения маршрутов при помощи протоколов маршрутизации IP. Хотя протоколы маршрутизации IP выполняют важную роль, термин "маршрутизация IP" относится только к процессу перенаправления пакетов.

Процесс маршрутизации IPv4

Несмотря на то что основы процесса маршрутизации уже упоминались в главе 4, в этом разделе повторим его этапы для справки. Для описания процесса используется много специфических терминов, обсуждавшихся в частях II и III. Ниже кратко рассматривается логика маршрутизации, чтобы гарантировать правильность понимания каждого этапа.

Процесс маршрутизации начинается с хоста, создающего пакет IP. Сначала хост решает вопрос: не принадлежит ли IP-адрес получателя этого нового пакета локальной подсети? Для определения диапазона адресов в локальной подсети хост использует собственный IP-адрес и маску. На основании собственных выводов о диапазоне адресов локальной подсети хост действует следующим образом.

Этапы перенаправления пакетов IP хостом

Ключевая тема

Этап 1 Если получатель локальный, передача осуществляется непосредственно:

A. MAC-адрес хоста получателя определяется при помощи уже существующей записи таблицы протокола преобразования адресов (ARP) или сообщения ARP, позволяющего изучить эту информацию.

B. Пакет IP инкапсулируется во фрейм канала связи с адресом канала связи *хоста получателя* (destination host)

Этап 2 Если получатель не является локальным, то передача осуществляется на стандартный шлюз:

A. MAC-адрес стандартного шлюза определяется при помощи уже существующей записи таблицы ARP или сообщения ARP, позволяющего изучить эту информацию

B. Пакет IP инкапсулируется во фрейм канала связи с адресом канала связи *стандартного шлюза* (default gateway)

Эти концепции представлены на рис. 16.1. Хост A на рисунке посылает пакет локальному хосту D непосредственно. Но для хоста B, расположенного с другой стороны маршрутизатора, а следовательно, в другой подсети, хост A посылает пакет на свой стандартный маршрутизатор (R1). (Термины *стандартный шлюз* (default gateway) и *стандартный маршрутизатор* (default router) — синонимы.)

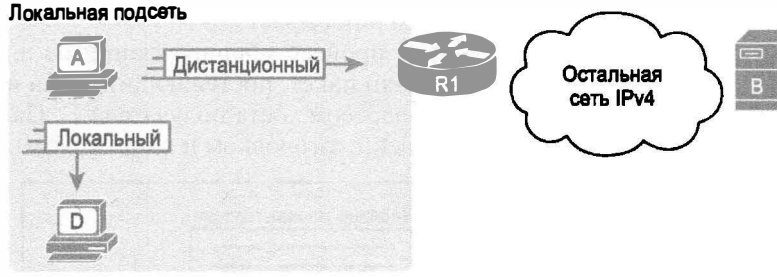


Рис. 16.1. Логика маршрутизации хоста

У маршрутизаторов немного больше работы при маршрутизации по сравнению с хостами. В то время как логика хоста начинается с пакета IP, находящегося в памяти, маршрутизатору, прежде чем дойти до того положения, необходимо проделать некоторую работу. Ниже приведено пять этапов логики маршрутизации, причем на первых двух этапах осуществляется только получение фрейма и извлечение пакета IP перед принятием решения об адресе получателя пакета на этапе 3.

Этапы перенаправления пакетов IP маршрутизатором

Ключевая тема

Этап 1 Для каждого полученного фрейма канала связи принимается решение, обрабатывать его или нет. Обрабатывается фрейм так:

A. Проверка фрейма на ошибки (по полю контрольной суммы фрейма (FCS) в концевике канала связи).

B. Адрес канала связи получателя фрейма — это адрес маршрутизатора (или соответствующий многоадресатный или широковещательный адрес)

Этап 2 Перед решением об обработке фрейма на этапе 1 он извлекается из фрейма канала связи

Этап 3 Принимается решение о маршрутизации. Для этого по IP-адресу получателя пакета осуществляется поиск соответствующего элемента таблицы маршрутизации, содержащего маршрут к получателю. Этот маршрут идентифицирует исходящий интерфейс маршрутизатора, а возможно, и следующий транзитный маршрутизатор

Этап 4 Помещает (инкапсулирует) пакет во фрейм канала связи, соответствующего исходящему интерфейсу. По мере необходимости для поиска MAC-адреса следующего устройства используется протокол ARP

Этап 5 Фрейм передается на исходящий интерфейс, указанный в соответствующем маршруте IP

ВНИМАНИЕ!

Тот факт, что этот список состоит из пяти этапов, а не другого количества, не имеет никакого значения. Значение имеют концепции каждого этапа, и именно они будут в экзаменационных вопросах. Нет никакой необходимости запоминать, какая часть логики с каким именно этапом связана.

Этапы процесса маршрутизации насчитывают много подробностей, но иногда его можно рассматривать упрощенно. Например, отбросив некоторые детали, этапы этого процесса можно пересказать следующим образом:

Маршрутизатор получает фрейм, извлекает из него пакет, решает, куда его перенаправить, помещает пакет в другой фрейм и посылает его.

Для лучшей демонстрации этих этапов процесс маршрутизации из пяти этапов представлен на рис. 16.2. На рисунке показан пакет, поступающий слева на входной интерфейс Ethernet маршрутизатора с IP-адресом хоста получателя C. Пакет поступает инкапсулированным во фрейм Ethernet (с заголовком и концевиком).

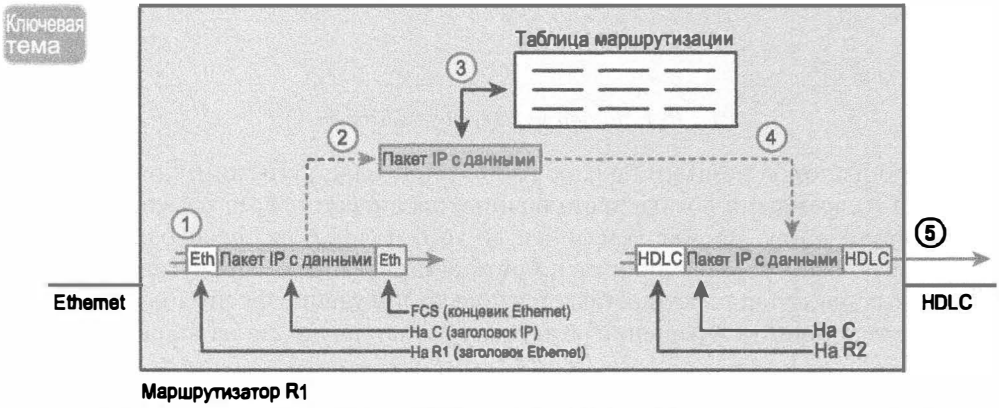


Рис. 16.2. Пять этапов маршрутизации, осуществляемых маршрутизатором

Маршрутизатор R1 обрабатывает фрейм и пакет, как показано цифрами на рисунке в соответствии с процессом из пяти этапов.

1. Маршрутизатор R1 отмечает, что полученный фрейм Ethernet прошел проверку FCS и что получатель Ethernet имеет MAC-адрес маршрутизатора R1, поэтому маршругизатору R1 предстоит обрабатывать фрейм.

2. Маршрутизатор R1 извлекает пакет IP из заголовка и концевика фрейма Ethernet.
3. Маршрутизатор R1 ищет IP-адрес получателя пакета IP в таблице маршрутизации IP.
4. Маршрутизатор R1 инкапсулирует пакет IP в новый фрейм канала связи (в данном случае в заголовок и концевик протокола HDLC).
5. Маршрутизатор R1 передает пакет IP в новом фрейме HDLC через последовательный канал связи справа.

ВНИМАНИЕ!

В этой главе пакет IP, инкапсулируемый во фрейме канального уровня, приведен на нескольких рисунках. На этих рисунках зачастую показан как заголовок канала связи, так и концевик с пакетом IP всередине. Все пакеты IP включают заголовок IP и инкапсулируемые в них данные.

Пример маршрутизации IP

Далее рассматриваются этапы маршрутизации через несколько устройств. В данном случае хост A (172.16.1.9) посылает пакет хосту B (172.16.2.9), используя логику маршрутизации хоста. Затем маршрутизатор R1 перенаправляет пакет согласно логике из пяти этапов.

На рис. 16.3 приведена типичная схема IP-адресации для сети IPv4 с типичными сокращениями адресов. Если отображать полные IP-адреса для каждого интерфейса маршрутизатора, схема окажется слишком загроможденной. По возможности на рисунках обычно указывают подсеть, затем последний (или два) октет IP-адресов, — этого вполне достаточно для идентификации IP-адреса без излишеств. Предположим, например, что хост использует IP-адрес 172.16.1.9 из подсети 172.16.1.0/24 (в которой все адреса начинаются с 172.16.1), поэтому около пиктограммы хоста A изображена цифра ".9". Вот другой пример: маршрутизатор R1 использует адрес 172.16.1.1 на своем интерфейсе LAN, адрес 172.16.4.1 на последовательном интерфейсе и адрес 172.16.5.1 на еще одном последовательном интерфейсе.

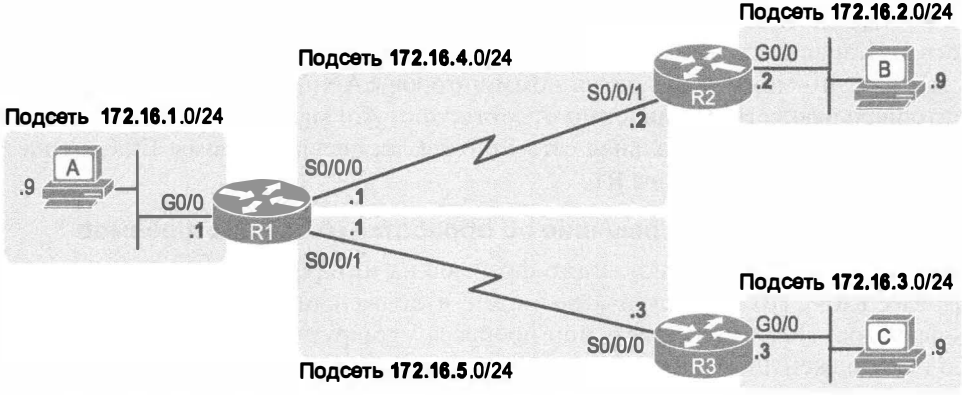


Рис. 16.3. Сеть IPv4, используемая в примере с пятью этапами маршрутизации

Теперь рассмотрим пример с хостом A (172.16.1.9), посылающим пакет хосту B (172.16.2.9).

Хост перенаправляет пакет IP на стандартный маршрутизатор (шлюз)

В этом примере хост A использует некое приложение, передающее данные хосту B (172.16.2.9). После формирования хостом A пакета IP в памяти логика хоста A сводится к следующему.

- Мой IP-адрес/маска — 172.16.1.9/24, следовательно, моя локальная подсеть содержит номера 172.16.1.0–172.16.1.255 (включая идентификаторы и широковещательные адреса подсети).
- Адрес получателя, 172.16.2.9, явно находится не в моей локальной подсети.
- Пошлю пакет на мой стандартный шлюз по адресу 172.16.1.1.
- Чтобы послать пакет, инкапсулирую его во фрейме Ethernet. MAC-адрес получателя будет принадлежать интерфейсу G0/0 маршрутизатора R1 (стандартный шлюз хоста A).

Эти концепции представлены на рис. 16.4: IP-адрес и MAC-адрес получателя во фрейме и пакете, посланном хостом A в данном случае.

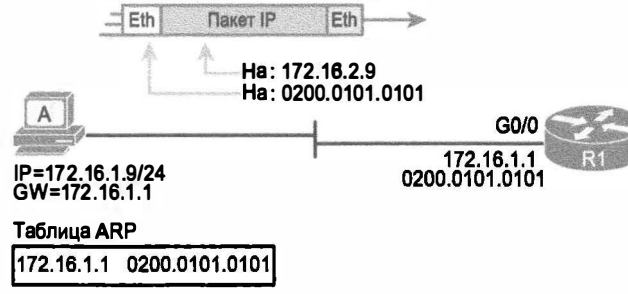


Рис. 16.4. Хост A посылает пакет хосту B

Обратите внимание на то, что каналы Ethernet LAN представлены на рисунке как простые линии, но они могут включать любые устройства, обсуждавшиеся в части II. Канал LAN может быть просто кабелем между хостом A и маршрутизатором R1, или это может быть сотня коммутаторов LAN, объединенных в огромную территориальную сеть. Независимо от этого, хост A и маршрутизатор R1 находятся в той же сети VLAN, и локальная сеть Ethernet доставляет фреймы Ethernet на интерфейс G0/0 маршрутизатора R1.

1-й этап маршрутизации: решение об обработке входящих фреймов

Маршрутизаторы получают много фреймов на интерфейсах, в частности на интерфейсах LAN. Но маршрутизатор может и должен игнорировать некоторые из этих фреймов. Поэтому первый этап процесса маршрутизации начинается с решения о том, должен ли маршрутизатор обработать фрейм или отбросить его.

Сначала маршрутизатор осуществляет простую, но очень важную проверку (этап 1A процесса), — он должен игнорировать все фреймы, переданные с ошибками. Для проверки фрейма на ошибки передачи маршрутизатор использует поле FCS заго-

ловка канала связи. (Маршрутизатор не предпринимает попыток восстановления после ошибок; т.е. не запрашивает повторную передачу данных.)

Маршрутизатор проверяет также адрес канала связи получателя (этап 1B), чтобы выяснить, предназначен ли фрейм для маршрутизатора. Например, фреймы, посланные на одноадресатный MAC-адрес интерфейса маршрутизатора, однозначно предназначались ему. Но маршрутизатор вполне может получить фрейм, посланный на некий другой одноадресатный MAC-адрес. Такой фрейм следует игнорировать.

Маршрутизаторы получат одноадресатные фреймы, посланные на другие устройства сети VLAN, благодаря принципу работы коммутаторов LAN. Помните, коммутаторы LAN рассылают одноадресатные фреймы с неизвестным получателем — это фреймы, для которых коммутатор не нашел MAC-адрес получателя в таблице MAC-адресов. Иногда маршрутизаторы получают фреймы, предназначенные для некоего другого устройства, причем с MAC-адресом другого устройства. Такие фреймы маршрутизаторы должны игнорировать.

В этом примере хост A посылает фрейм на MAC-адрес маршрутизатора R1. Таким образом, после получения этого фрейма и проверки его FCS, подтверждающей отсутствие ошибки, маршрутизатор R1 устанавливает, что фрейм предназначен для MAC-адреса маршрутизатора R1 (в данном случае 0200.0101.0101). Поскольку все проверки пройдены, маршрутизатор R1 решает обработать фрейм, как показано на рис. 16.5. (Обратите внимание на большой прямоугольник на рисунке, он представляет внутреннюю организацию маршрутизатора R1.)

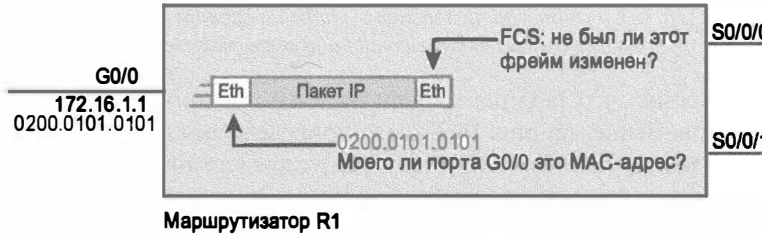


Рис. 16.5. 1-й этап маршрутизации: проверка FCS и MAC-адреса получателя

2-й этап маршрутизации: извлечение пакета IP

Выяснив, что полученный фрейм следует обработать (этап 1), маршрутизатор предпринимает следующий шаг — извлекает пакет. В памяти маршрутизатора не нужен ни заголовок, ни концевик канала связи первоначального фрейма, поэтому маршрутизатор удаляет их, оставляя только пакет IP, как показано на рис. 16.6. Обратите внимание, что IP-адрес получателя остается неизменным (172.16.2.9).

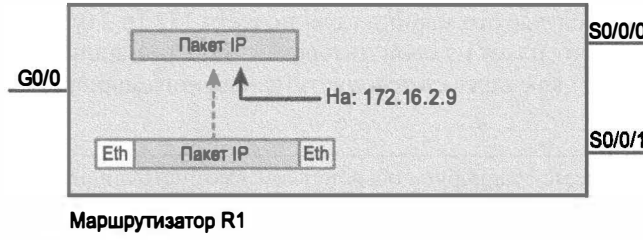


Рис. 16.6. 2-й этап маршрутизации: извлечение пакета

3-й этап маршрутизации: выбор направления перенаправления пакета

Второй этап маршрутизации несложен, в отличие от этапа 3. Теперь маршрутизатор должен выбрать направление перенаправления пакетов. Для этого используется таблица маршрутизации IP маршрутизатора и логика соответствия при поиске адреса получателя пакета в таблице.

Таблица маршрутизации IP содержит несколько записей маршрутов. Каждая запись маршрута содержит несколько фактов, которые в свою очередь могут быть сгруппированы, как на рис. 16.7. Часть записи используется для поиска соответствия адресу получателя пакета, в то время как остальная часть записи содержит инструкцию по перенаправлению, т.е. куда послать пакет.

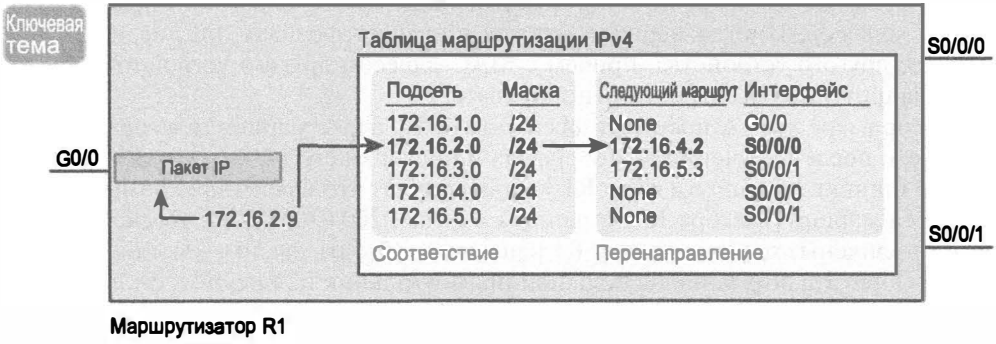


Рис. 16.7. 3-й этап маршрутизации: таблица маршрутизации IP имеет части соответствия и перенаправления

Обратите внимание, что таблица маршрутизации в данном случае содержит пять записей маршрутов. Выше, на рис. 16.3, была приведена вся сеть с пятью подсетями. Таким образом, у маршрутизатора R1 есть маршрут для каждой из этих пяти подсетей.

Теперь рассмотрим ту часть записей маршрутов, которую маршрутизатор R1 будет использовать для поиска соответствия пакету. Для полного определения подсети каждая запись маршрута содержит идентификатор и маску подсети. Маршрутизатор ищет соответствие IP-адреса получателя пакета (172.16.2.9) в таблице маршрутизации, сравнивая его с диапазоном адресов, определенных каждой подсетью. Точнее, маршрутизатор просматривает информацию о подсети и маске, для которой достаточно применить несколько математических действий, чтобы выяснить, в какой из этих подсетей располагается адрес 172.16.2.9 (в подсети 172.16.2.0/24).

И наконец, обратимся к правой части рисунка — к инструкциям перенаправления для этих пяти маршрутов. После того как маршрутизатор найдет соответствующий маршрут, он использует информацию о перенаправлении, чтобы узнать, куда послать пакет далее. В данном случае это маршрут для подсети 172.16.2.0/24, поэтому маршрутизатор R1 перенаправит пакет на свой интерфейс S0/0/0, маршрутизатору R2, указав его IP-адрес (172.16.4.2) как адрес следующего транзитного маршрутизатора.

ВНИМАНИЕ!

Для дистанционных подсетей маршруты обычно указывают исходящий интерфейс и IP-адрес следующего транзитного маршрутизатора. Для подсетей, подключенных к маршрутизатору непосредственно, указывают только исходящий интерфейс, поскольку пакеты для этих получателей уже не будут посланы на другой маршрутизатор.

4-й этап маршрутизации: инкапсуляция пакета в новый фрейм

Теперь маршрутизатор знает, куда перенаправить пакет. Но маршрутизаторы не могут перенаправить пакет без оболочки из заголовка и концевика канала связи (инкапсуляция).

Инкапсуляция пакетов для последовательных каналов связи не требует особого размышления из-за простоты протоколов PPP и HDLC. Как упоминалось в главе 3, поскольку последовательные каналы соединяют только два устройства (отправителя и получателя), адресация канала связи не имеет значения. В данном примере маршрутизатор R1 перенаправляет пакет через интерфейс S0/0/0, поместив его во фрейм HDLC, как показано на рис. 16.8.

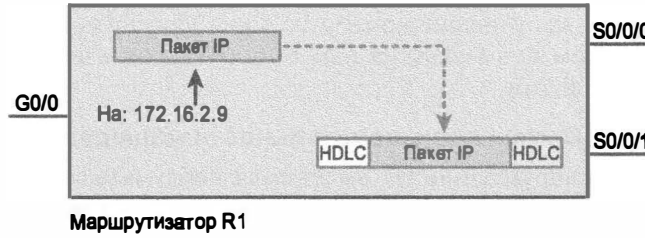


Рис. 16.8. 4-й этап маршрутизации: инкапсуляция пакета

Отметим, что при некоторых других типах каналов связи у маршрутизатора будет побольше работы на этом этапе маршрутизации. Например, иногда маршрутизатор перенаправляет пакеты на интерфейс Ethernet. Чтобы инкапсулировать пакет IP, маршрутизатор должен создать заголовок Ethernet, включающий правильное значение MAC-адреса получателя.

Рассмотрим, например, другую типовую сеть: с каналом связи Ethernet WAN между маршрутизаторами R1 и R2. Маршрутизатор R1 выбирает маршрут, указывающий перенаправить пакет на интерфейс Ethernet G0/1 для маршрутизатора R2 (172.16.6.2). Для этого маршрутизатор R1 должен поместить в заголовок MAC-адрес маршрутизатора R2. Чтобы сделать это, он использует информацию из таблицы ARP, как показано на рис. 16.9. Если у маршрутизатора R1 нет в таблице ARP записи для адреса 172.16.6.2, то он использует протокол ARP для изучения этого MAC-адреса.

Рис. 16.9. 4-й этап маршрутизации: маршрутизатор R1 с исходящим интерфейсом LAN

5-й этап маршрутизации: передача фрейма

После завершения подготовки фрейма маршрутизатору остается только передать его. Возможно, маршрутизатору придется подождать, особенно если другие фреймы уже ждут своей очереди для отправки через интерфейс.

Внутренняя обработка на маршрутизаторах Cisco

До сих пор в этой главе обсуждалась только одна точка зрения на внутреннюю работу хоста и маршрутизатора. Но чтобы маршрутизаторы Cisco были конкурентоспособны, они должны осуществлять процесс маршрутизации хорошо, быстро и во всех видах сред. В противном случае конкуренты Cisco могли бы утверждать, что их маршрутизаторы работают лучше и способны перенаправлять больше пакетов за секунду, что повредит бизнесу компании Cisco.

В данном разделе немного подробнее рассматривается то, как маршрутизаторы Cisco фактически реализуют перенаправление IP. До сих пор в этой главе обсуждение было довольно общим и соответствовало раннему типу внутренних процессов на маршрутизаторах Cisco — *процессорной коммутации* (process switching). Здесь обсуждаются проблемы, вынудившие компанию Cisco улучшить внутренний процесс маршрутизации при том же результате: пакет поступает на маршрутизатор в одном фрейме, а покидает в другом.

Потенциальные проблемы производительности маршрутизации

Изучая процесс маршрутизации IP, имеет смысл обдумать все его частности, обсуждаемые ниже. Таким образом, маршрутизаторы тратят некоторое время на выполнение действий по перенаправлению отдельного пакета IP. Фактически даже очень медленные маршрутизаторы должны перенаправлять десятки тысяч пакетов в секунду; поэтому они не могут нести больших затрат на обработку каждого пакета.

Процесс поиска в таблице маршрутизации соответствия адресу получателя пакета IP фактически может занять много процессорного времени. В примере на рис. 16.7 приведено только пять маршрутов, но в корпоративных сетях обычно есть тысячи маршрутов IP, а у маршрутизаторов ядра Интернета их сотни тысяч. Теперь подумайте о процессоре маршрутизатора, ведь он должен осуществлять поиск в списке из 100 000 записей для каждого пакета, а их следует перенаправлять сотни тысяч в секунду! А что если маршрутизатору пришлось бы еще вычислять каждый раз подсети, диапазоны адресов в каждой подсети для каждого маршрута? Эти вычисления отняли бы слишком много мощности процессора.

За последние годы компания Cisco выработала несколько способов оптимизации внутреннего процесса перенаправления пакетов. Некоторые из них связаны со специфической моделью или серией маршрутизаторов. Коммутаторы уровня 3 осуществляют перенаправление в *специализированных интегральных микросхемах* (Application Specific Integrated Circuits — ASIC), специально созданных для перенаправления фреймов и пакетов. Базовая логика соответствует приведенному ранее списку из пяти этапов, а оптимизация осуществляется в зависимости от аппаратных средств маршрутизатора и его программного обеспечения так, чтобы снизить нагрузку на процессор и сократить дополнительные затраты на перенаправление пакетов IP.

Быстрая коммутация маршрутизаторов Cisco и CEF

Исторически сложилось так, что у компании Cisco было три главных варианта внутренней логики маршрутизации во всех семействах маршрутизаторов. Первоначально, в конце 1980–начале 1990-х, на первых маршрутизаторах Cisco использовалась внутренняя логика, известная как *процессорная коммутация*. Процессорная коммутация работает в основном так, как было описано до сих пор в этой главе, без дополнительной оптимизации.

Затем, в начале 1990-х, компания Cisco применила дополнительную внутреннюю логику маршрутизации — *быструю коммутацию* (fast switching). Быстрая коммутация несколько оптимизирована по сравнению с прежней логикой процессорной коммутации. В первую очередь, в дополнение к таблице маршрутизации, она хранит другой список, содержащий индивидуальные IP-адреса для недавно перенаправленных пакетов. Этот кеш быстрой коммутации хранит также копии новых заголовков канала связи, используемых при перенаправлении пакетов каждому получателю. Таким образом, вместо создания нового заголовка канала связи для каждого пакета, предназначенного для конкретного IP-адреса, маршрутизатор просто копирует прежний заголовок.

В конце 1990-х компания Cisco усовершенствовала быструю коммутацию до логики CEF (Cisco Express Forwarding). Как и быстрая коммутация, CEF использует дополнительные таблицы для ускорения поиска и хранит исходящие заголовки канала связи. Однако логика CEF организует свои таблицы маршрутизации для всех получателей, а не только для некоторых конкретных IP-адресов получателя. Логика CEF использует также более сложные алгоритмы поиска и двоичные древовидные структуры по сравнению с логикой быстрой коммутации. В результате поиск в таблице CEF занимает существенно меньше времени, чем даже при быстрой коммутацией. Заголовки канала связи здесь также кешируются.

Современные модели маршрутизаторов Cisco и версии операционной системы IOS используют логику CEF. В табл. 16.1 кратко сравниваются процессорная коммутация, быстрая коммутация и CEF.

Таблица 16.1. Сравнение логики коммутации пакетов

Улучшает эффективность маршрутизации ...	Процессорная коммутация	Быстрая коммутация	CEF
... сохраняя заголовки канала связи для инкапсуляции пакетов	Нет	Да	Да
... используя другие таблицы, с более быстрым поиском, перед обращением к таблице маршрутизации	Нет	Да	Да
... организует таблицы, используя древовидные структуры для очень быстрого поиска и сокращения времени перенаправления пакетов	Нет	Нет	Да

Настройка подключенных маршрутов

Изначально протокол IPv4 включен на маршрутизаторах Cisco глобально. Чтобы маршрутизатор был готов перенаправлять пакеты на конкретном интерфейсе, следует настроить на нем IP-адрес и добиться для него состояния `up/up`. Только в этом случае маршрутизаторы смогут перенаправлять пакеты IP на специфический интерфейс.

После того как маршрутизатор сможет перенаправлять пакеты IP через один или несколько интерфейсов, ему понадобятся маршруты. Маршрутизаторы могут добавлять маршруты в свои таблицы маршрутизации тремя способами.

Три источника маршрутов IP для маршрутизаторов

Ключевая тема

- *Подключенные маршруты* (connected route). Добавляются подкомандой интерфейса `ip address` на локальном маршрутизаторе.
- *Статические маршруты* (static route). Добавляются глобальной командой конфигурации `ip route` на локальном маршрутизаторе.

- *Протоколы маршрутизации* (routing protocol). Дополнительная функция настройки на всех маршрутизаторах, обеспечивающая динамический обмен данными о сети между маршрутизаторами, позволяющая им изучить все маршруты.

Во втором из трех разделов этой главы обсуждается несколько вариантов настройки подключенных маршрутов, в то время как в последнем разделе обсуждаются статические маршруты.

Подключенные маршруты и команда `ip address`

Маршрутизатор Cisco автоматически добавляет в свою таблицу маршрутизации маршруты для подсетей, подключенных к каждому его интерфейсу, с учетом истинности следующих двух фактов.

Ключевая тема

Правила создания маршрутизатором подключенного маршрута

- Интерфейс находится в рабочем состоянии, т.е. в выводе команды `show interfaces` состояние интерфейса для линии `up` и для протокола `up`.
- Интерфейсу присвоен IP-адрес подкомандой интерфейса `ip address`.

Концепция подключенных маршрутов относительно проста. Маршрутизатор, конечно, должен знать номер подсети, физически соединенной с каждым ее интерфейсом, чтобы перенаправлять в нее пакеты. Маршрутизатор может просто вычислить идентификатор подсети по IP-адресу интерфейса и маске. Однако этот маршрут нужен маршрутизатору только тогда, когда интерфейс включен и работает, поэтому маршрутизатор включает подключенный маршрут в таблицу маршрутизации, только когда интерфейс работает.

Пример 16.1 и рис. 16.10 демонстрируют подключенные маршруты на маршрутизаторе R1. Рисунок повторяет ту же сеть IP, что и ранее. Первая часть примера демонстрирует конфигурацию IP-адресов на всех трех интерфейсах маршрутизатора R1. В конце примера приведен вывод команды `show ip route`, перечисляющей эти маршруты с кодом маршрута `c`, означающим `connected` (подключенный).

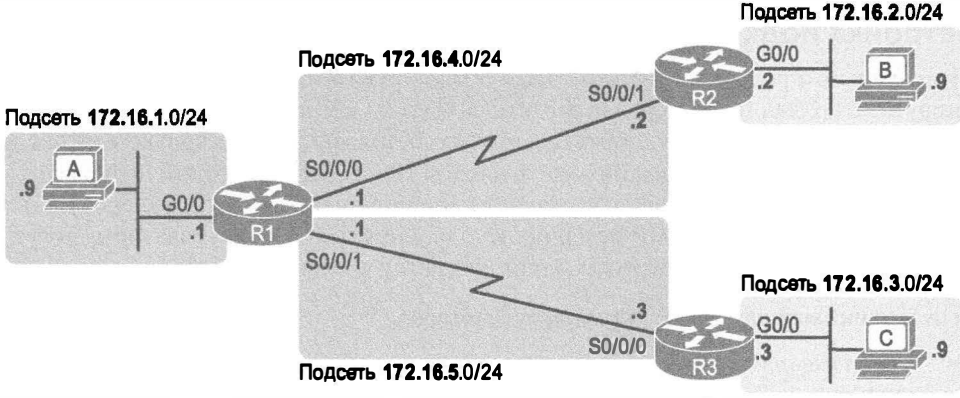


Рис. 16.10. Пример сети, демонстрирующий подключенные маршруты

Пример 16.1. Подключенные и локальные маршруты на маршрутизаторе R1

```
! Отрывок из файла running-config...
!
interface GigabitEthernet0/0
  ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/0
  ip address 172.16.4.1 255.255.255.0
!
interface Serial0/0/1
  ip address 172.16.5.1 255.255.255.0

R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1,
       L2 - IS-IS level-2, ia - IS-IS inter area, * - candidate default,
       U - per-user static route, o - ODR, P - periodic downloaded static
       route, H - NHRP, l - LISP, + - replicated route, % - next hop
       override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
C     172.16.4.0/24 is directly connected, Serial0/0/0
L     172.16.4.1/32 is directly connected, Serial0/0/0
C     172.16.5.0/24 is directly connected, Serial0/0/1
L     172.16.5.1/32 is directly connected, Serial0/0/1
```

Рассмотрим подробно каждый из трех маршрутов в выводе команды `show ip route`. Каждый предваряется буквой `C` в первом столбце, и у каждого есть комментарий "directly connected" (подключен непосредственно); оба идентифицируют маршрут как подключенный к маршрутизатору. В начальной части каждого маршрута указаны его параметры (идентификатор подсети и маска), как показано в примере на рис. 16.7. В конце каждого из этих маршрутов приведен исходящий интерфейс.

Обратите также внимание, что маршрутизатор автоматически создает маршруты другого вида — *локальные маршруты* (local route). Они определяют маршрут для одного конкретного IP-адреса, заданного на интерфейсе маршрутизатора. У каждого локального маршрута есть префикс длиной /32, определяющий *маршрут хоста* (host route), т.е. маршрут только для этого одного IP-адреса. Например, последний локальный маршрут, 172.16.5.1/32, соответствует только IP-адресу 172.16.5.1. Маршрутизаторы используют локальные маршруты, обладающие собственными локальными IP-адресами, чтобы эффективней перенаправлять пакеты, посланные на сам маршрутизатор.

После того как маршрутизатор добавит подключенные маршруты, он сможет перенаправлять пакеты IPv4 между этими подсетями.

Маршрутизация между подсетями VLAN

Почти все корпоративные сети используют виртуальные локальные сети (VLAN). Для перенаправления пакетов IP в и из сетей VLAN (точнее, подсетей, находящихся

в каждой из них) маршрутизатор должен иметь IP-адрес каждой подсети и подключенный маршрут к каждой из этих подсетей. Хосты в каждой подсети могут использовать IP-адреса маршрутизатора как адрес своего стандартного шлюза.

Для соединения маршрутизатора с каждой подсетью в сети VLAN есть три возможности. Однако первая из них требует слишком много интерфейсов и каналов связи, поэтому она упоминается только для полноты списка.

Ключевая тема

Три возможности соединения маршрутизаторов с каждой сетью VLAN

- С коммутатором каждой сети VLAN соединен один кабель и интерфейс LAN маршрутизатора (обычно не используется).
- Маршрутизатор соединен с коммутатором LAN магистральным каналом VLAN.
- Используется коммутатор уровня 3.

На рис. 16.11 приведен пример сети, использующей вторую и третью возможности. Здесь слева представлена центральная площадка территориальной сети LAN с 12-ю сетями VLAN. На центральной площадке два коммутатора уровня 3 объединяют функции маршрутизатора и коммутатора, осуществляя маршрутизацию между всеми 12 подсетями (или VLAN). Дистанционные ветви площадки (с правой стороны рисунка) используют по две VLAN; для подключения и маршрутизации в обеих сетях VLAN каждый маршрутизатор использует магистральный канал VLAN.

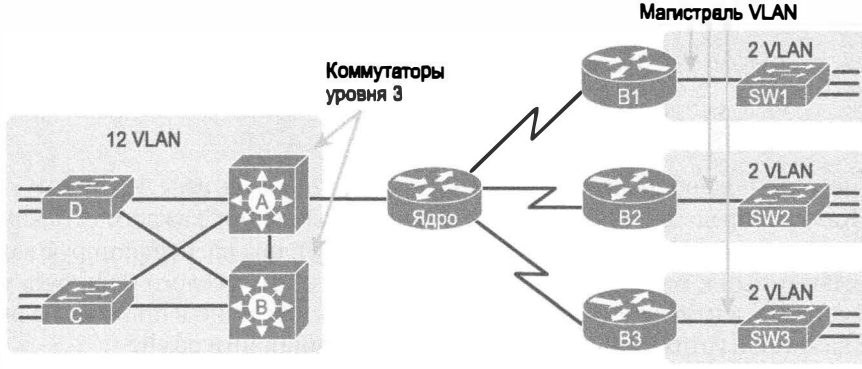


Рис. 16.11. Субинтерфейсы на маршрутизаторе R1

Обратите внимание: рис. 16.11 — это только пример. Инженер может использовать коммутацию третьего уровня на каждой площадке или маршрутизаторы с магистральным соединением VLAN на каждой площадке. Поскольку эта глава посвящена в основном подробностям настройки, рассмотрим ее на нескольких следующих страницах.

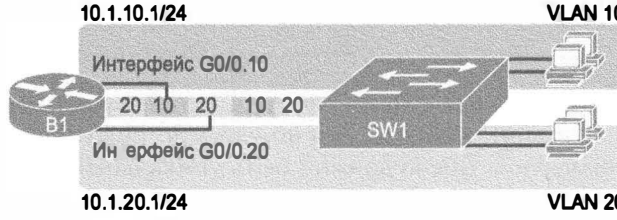
Настройка маршрутов к VLAN на маршрутизаторах 802.1Q

В этом разделе обсуждается перенаправление пакетов в подсети, ассоциируемые с сетями VLAN, подключенными к маршрутизатору 802.1Q магистральным каналом. Это довольно длинное описание неудобно повторять каждый раз, поэтому со временем был прият более короткий термин — *маршрутизатор на палочке* (Router On A Stick — ROAS).

Схема сети ROAS подразумевает использование конфигурации магистрального соединения VLAN маршрутизатора для предоставления маршрутизатору логического

интерфейса, соединенного с каждой сетью VLAN, а потому каждая подсеть находится в отдельной VLAN. В основе такой магистральной конфигурации лежит концепция *субинтерфейсов* (subinterface). Для каждой сети VLAN на магистральном канале у маршрутизатора должен быть IP-адрес и маска. Но маршрутизатор использует только один физический интерфейс, настроенный командой `ip address`. Компания Cisco решает эту проблему за счет создания нескольких виртуальных интерфейсов маршрутизатора, ассоциируемых с каждой сетью VLAN на данном магистральном канале (по крайней мере, для каждой сети VLAN, поддерживаемой магистральным каналом). Компания Cisco называет эти виртуальные интерфейсы *субинтерфейсами*.

Конфигурация ROAS создает субинтерфейс для каждой сети VLAN на магистральном канале, а маршрутизатор затем обрабатывает все фреймы, отмеченные соответствующим идентификатором VLAN, как будто они были отправлены или приняты этим субинтерфейсом. На рис. 16.12 концепция представлена на примере маршрутизатора B1, одного из маршрутизаторов ветви на рис. 16.11. Поскольку этот маршрутизатор должен перенаправлять пакеты только между двумя сетями VLAN, на рисунке представлены также два субинтерфейса, G0/0.10 и G0/0.20, создающих новую область в конфигурации, где могут быть заданы параметры конфигурации каждой VLAN. Маршрутизатор рассматривает фреймы, отмеченные сетью VLAN 10, как будто проходящими через интерфейс G0/0.10, а фреймы, отмеченные сетью VLAN 20, как будто проходяшими через интерфейс G0/0.20.



Ключевая тема

Рис. 16.12. Субинтерфейсы на маршрутизаторе B1

Обратите также внимание, что маршрутизаторы Cisco не пытаются договариваться о магистральном соединении, поэтому магистральное соединение маршрутизатора и коммутатора следует настроить вручную. В данной главе обсуждается сторона маршрутизатора магистральной конфигурации; соответствующий интерфейс коммутатора должен быть настроен командой `switchport mode trunk`.

Пример 16.2 демонстрирует полную конфигурация магистрали 802.1Q маршрутизатора B1 в соответствии с рис. 16.12. Этапы настройки конфигурации магистрального соединения 802.1Q на маршрутизаторе таковы.

Этапы настройки магистрального соединения 802.1Q

Ключевая тема

Этап 1 Создать индивидуальный субинтерфейс для каждой маршрутизируемой сети VLAN (`interface` *тип номер_субинтерфейса*)

Этап 2 Включить протокол 802.1Q и ассоциировать конкретную сеть VLAN с субинтерфейсом в режиме конфигурации субинтерфейса (`encapsulation dot1q` *идентификатор_vlan*)

Этап 3 Настроить параметры IP (адрес и маску) в режиме конфигурации субинтерфейса (`ip address` *адрес маска*)

Пример 16.2. Конфигурация маршрутизатора для магистрали 802.1Q согласно рис. 16.12

```
B1# show running-config
! Показаны лишь некоторые строки
interface gigabitethernet 0/0
! No IP address up here! No encapsulation up here!
!
interface gigabitethernet 0/0.10
  encapsulation dot1q 10
  ip address 10.1.10.1 255.255.255.0
!
interface gigabitethernet 0/0.20
  encapsulation dot1q 20
  ip address 10.1.20.1 255.255.255.0
!
B1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
! Строки опущены для краткости

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C     10.1.10.0/24 is directly connected, GigabitEthernet0/0.10
L     10.1.10.1/32 is directly connected, GigabitEthernet0/0.10
C     10.1.20.0/24 is directly connected, GigabitEthernet0/0.20
L     10.1.20.1/32 is directly connected, GigabitEthernet0/0.20
```

Сначала рассмотрим номера субинтерфейсов. Номер субинтерфейса начинается с точки, как .10 и .20 в данном случае. Эти номера могут быть любыми числами от 1 до более чем 4 миллиардов. Число должно быть уникальным среди всех субинтерфейсов, связанных с данным физическим интерфейсом. Фактически номер субинтерфейса не должен даже соответствовать идентификатору своей VLAN. (Команда `encapsulation` без номера субинтерфейса создаст идентификатор VLAN, ассоциированный с субинтерфейсом.)

ВНИМАНИЕ!

Хоть и не обязательно, только во избежание беспорядка, на большинстве площадок номера субинтерфейсов выбирают в соответствии с идентификаторами VLAN, как показано в примере 16.2.

Каждая конфигурация субинтерфейса насчитывает две подкоманды. Одна (`encapsulation`) разрешает магистральное соединение и определяет сеть VLAN, фреймы которой будут пересекать субинтерфейс. Команда `ip address` работает точно тот же, как и на любом другом интерфейсе. Обратите внимание, если физический интерфейс Ethernet находится в состоянии `up/up`, то субинтерфейс также находится в этом состоянии и позволяет маршрутизатору добавлять подключенные маршруты, представленные в нижней части примера.

Теперь, когда у маршрутизатора есть рабочий интерфейс с настроенными IPv4-адресами, он может перенаправлять пакеты IPv4 на этих субинтерфейсах. Таким образом, маршрутизатор рассматривает эти субинтерфейсы как любой физический

интерфейс с точки зрения добавления подключенных маршрутов, распознавания этих маршрутов и перенаправления пакетов на связанные с ними подсети и из них.

ВНИМАНИЕ!

Несмотря на то что пример 16.2 демонстрируют конфигурацию по протоколу 802.1Q, конфигурация по протоколу ISL на том же маршрутизаторе была бы фактически той же. Только в каждом случае вместо ключевого слова `dot1q` использовалось бы слово `isl`.

Последовательность настройки собственной сети VLAN 802.1

- Введите команду `ip address` на физическом интерфейсе, но без команды `encapsulation` (маршрутизатор полагает, что этот физический интерфейс использует собственную сеть VLAN).
- Введите команду `ip address` на субинтерфейсе и подкоманду `encapsulation...native`.

Пример 16.3 демонстрирует обе возможности настройки при небольших изменениях той же конфигурации в примере 16.2. В данном случае сеть VLAN 10 становится собственной сетью VLAN. Верхняя часть примера демонстрирует возможность настройки маршрутизатора на использование собственной сети VLAN 10 с учетом, что коммутатор также был настроен на использование собственной сети VLAN 10. Вторая половина примера демонстрирует настройку той же собственной сети VLAN на субинтерфейсе.

Пример 16.3. Настройка использования собственной сети VLAN 10 на маршрутизаторе B1

```
! Первая возможность: задать IP-адрес собственной сети VLAN
! на физическом интерфейсе
interface gigabitethernet 0/0
ip address 10.1.10.1 255.255.255.0
!
interface gigabitethernet 0/0.20
encapsulation dot1q 20
ip address 10.1.20.1 255.255.255.0

! Вторая возможность: как обычно, но с ключевым словом native
interface gigabitethernet 0/0.10
encapsulation dot1q 10 native
ip address 10.1.10.1 255.255.255.0
!
interface gigabitethernet 0/0.20
encapsulation dot1q 20
ip address 10.1.20.1 255.255.255.0
```

Кроме демонстрации конфигурации, команда `show vlans` на маршрутизаторе подробно объясняет использование магистральных интерфейсов маршрутизатора, какая из VLAN является собственной, а также немного статистики пакетов. Пример 16.4 демонстрирует вывод на основании конфигурации маршрутизатора B1 в примере 16.3 (нижняя часть), в которой собственная сеть VLAN 10 настраивается на субинтерфейсе G0/0.10. Вывод свидетельствует, что сеть VLAN 1 ассоциирована с физическим ин-

терфейсом, VLAN 10 — собственная сеть VLAN, ассоциированная с субинтерфейсом G0/0.10, а VLAN 20 — с субинтерфейсом G0/0.20.

Пример 16.4. Пример команды `show vlans` для типичной конфигурации магистрального соединения маршрутизатора

```
R1# show vlans

Virtual LAN ID: 1 (IEEE 802.1Q Encapsulation)

  vLAN Trunk Interface: GigabitEthernet0/0

Protocols Configured:  Address:  Received:  Transmitted:
Other                                 0            83
  69 packets, 20914 bytes input
  147 packets, 11841 bytes output

Virtual LAN ID: 10 (IEEE 802.1Q Encapsulation)

  vLAN Trunk Interface: GigabitEthernet0/0.10

  This is configured as native Vlan for the following interface(s) :
GigabitEthernet0/0
Protocols Configured:  Address:  Received:  Transmitted:
      IP               10.1.10.1        2             3
   Other                                  0             1

3 packets, 722 bytes input
4 packets, 264 bytes output

Virtual LAN ID: 20 (IEEE 802.1Q Encapsulation)

  vLAN Trunk Interface: GigabitEthernet0/0.20

Protocols Configured:  Address:  Received:  Transmitted:
      IP               10.1.20.1        0           134
   Other                                  0             1

0 packets, 0 bytes input
135 packets, 10498 bytes output
```

И наконец, тем, кто внимательно следит за экзаменационными темами, стоит обратить внимание на то, что подразделы о маршрутизации между VLAN и ROAS упоминаются как "восходящая маршрутизация" ("upstream routing"). Это очень простая концепция: использующий ROAS маршрутизатор может перенаправлять пакеты только между заданными подсетями магистрального канала VLAN, но более чем вероятно, что маршрутизатор должен будет также перенаправлять пакеты другим подсетям в других частях корпоративной сети. Тема "восходящий маршрутизации" просто напоминает о необходимости настроить другие интерфейсы маршрутизатора, статические маршруты, протокол маршрутизации и т.д. так, чтобы маршрутизатор мог перенаправить пакеты и другим получателям по "восходящей".

Настройка маршрутов к VLAN с использованием коммутаторов уровня 3

Еще одна возможность перенаправления трафика VLAN подразумевает использование такого устройства, как коммутатор уровня 3, или многоуровневого комму-

татора. Как упоминалось в главе 9, коммутатор уровня 3 — это единое устройство, выполняющее две основные функции: коммутацию LAN уровня 2 и маршрутизацию IP уровня 3. Коммутация LAN уровня 2 перенаправляет фреймы в каждой сети VLAN, но не между ними. Логика перенаправления уровня 3 (маршрутизация) обеспечивает передачу пакетов IP между сетями VLAN.

Настройка коммутатора уровня 3 очень похожа на настройку коммутатора уровня 2, представленную ранее в части II этой книги, лишь с небольшим добавлением для функций уровня 3. Функция коммутации уровня 3 нуждается в виртуальном интерфейсе, подключенном к каждой сети VLAN внутренне на коммутаторе. Эти *интерфейсы VLAN* (VLAN interface) действуют как интерфейсы маршрутизатора, обладая IP-адресом и маской. У коммутатора уровня 3 есть таблица маршрутизации IP с подключенными маршрутами от каждого из этих интерфейсов VLAN. (Эти интерфейсы называют также *коммутируемыми виртуальными интерфейсами* (Switched Virtual Interface — SVI).)

Для демонстрации этой концепции на рис. 16.13 представлена видоизмененная схема сети филиала, используемого на рис. 16.11 и 16.12. Рисунок демонстрирует функцию коммутатора уровня 3 с пиктограммой маршрутизатора в коммутаторе, подчеркивающей, что коммутатор перенаправляет пакеты. У ветви есть еще две пользовательские сети VLAN, поэтому коммутатор уровня 3 нуждается в одном интерфейсе VLAN для каждой сети VLAN. Кроме того, для доступа к WAN трафик должен еще поступать на маршрутизатор, поэтому коммутатор использует третью сеть VLAN (VLAN 30 в данном случае) для канала связи с маршрутизатором B1. Этот канал связи должен быть не магистралью, а каналом связи.

Рис. 16.13. Маршрутизация на интерфейсах VLAN коммутатора уровня 3

Ниже приведена последовательность настройки коммутации третьего уровня. Обратите внимание: на некоторых коммутаторах, таких как используемый в примерах данной книги коммутатор 2960, сначала необходимо разрешить маршрутизацию пакетов IPv4, а затем перезагрузить его. Остальная часть этапов (после этапа 1) применима ко всем моделям коммутаторов Cisco, поддерживающим коммутацию третьего уровня.

Настройка коммутации третьего уровня

Ключевая тема

Этап 1 Разрешите аппаратную поддержку маршрутизации IPv4. На коммутаторах 2960, например, используется глобальная команда конфигурации `sdm prefer lanbase-routing` и перезагрузка коммутатора

Этап 2 Разрешите маршрутизацию IPv4 глобально (команда `ip routing`)

Этап 3 Создайте интерфейсы VLAN по каждой сети VLAN, для которой коммутатор уровня 3 перенаправляет пакеты (команда `interface vlan` *идентификатор_vlan*)

Этап 4 Задайте на интерфейсе VLAN IP-адрес и маску (команда `ip address` *адрес маска* в режиме конфигурации интерфейса для данного интерфейса)

Этап 5 Если изначально интерфейс VLAN коммутатора отключен, включите его (команда `no shutdown`)

Пример 16.5 демонстрирует конфигурацию в соответствии с рис. 16.13. В данном случае на коммутаторе SW1 модели 2960 уже введена глобальная команда `sdm prefer lanbase-routing`, и он перезагружен. Пример демонстрирует настройку на всех трех интерфейсах VLAN.

Пример 16.5. Настройка интерфейса VLAN для коммутации третьего уровня

```
ip routing
!
interface vlan 10
  ip address 10.1.10.1 255.255.255.0
!
interface vlan 20
  ip address 10.1.20.1 255.255.255.0
!
interface vlan 30
  ip address 10.1.30.1 255.255.255.0
```

При этой конфигурации VLAN коммутатор готов перенаправлять пакеты между сетями VLAN, показанными на рис. 16.13. Для обеспечения маршрутизации пакетов коммутатор добавляет подключенные маршруты IP, как показано в примере 16.6; обратите внимание, что каждый маршрут отображается как подключенный к собственному интерфейсу VLAN.

Пример 16.6. Подключенные маршруты на коммутаторе уровня 3

```
SW1# show ip route
! легенда опущена для краткости

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C     10.1.10.0/24 is directly connected, Vlan10
L     10.1.10.1/32 is directly connected, Vlan10
C     10.1.20.0/24 is directly connected, Vlan20
L     10.1.20.1/32 is directly connected, Vlan20
C     10.1.30.0/24 is directly connected, Vlan30
L     10.1.30.1/32 is directly connected, Vlan30
```

Коммутатор нуждается также в дополнительных маршрутах к остальной части сети, представленной на рис. 16.11 (возможно использование статических маршрутов, как уже обсуждалось в этой главе).

Вторичная IP-адресация

Большинство современных сетей использует либо маршрутизаторы с магистральными каналами VLAN, либо коммутаторы уровня 3. В этом разделе рассматривается интересная, но малоиспользуемая функция, помогающая преодолеть некоторые затруднения в сети IP.

Предположим, что схема IP-адресации сети спланирована заранее. Впоследствии некая подсеть разрослась, и все ее допустимые IP-адреса были использованы. Что делать? Есть три возможности.

- Увеличить существующую подсеть, выбрав маску с бо́льшим количеством битов хоста. Существующие хосты должны изменить свои параметры маски подсети, а новые хосты могут использовать IP-адреса из расширенного интервала адресов.
- Перейти на совершенно новую (бо́льшую) подсеть. Все существующие устройства изменяют свои IP-адреса.
- Добавить вторую подсеть в той же области, используя вторичную адресацию.

Первая возможность хороша, пока новая подсеть не накладывается на существующие подсети. Например, если в проекте использовалась подсеть 172.16.2.0/24, и она исчерпала адреса, то инженер мог бы попытаться использовать маску /23. Получится подсеть 172.16.2.0/23 с диапазоном адресов от 172.16.2.1 до 172.16.3.254. Но если бы некой другой части сети уже была присвоена подсеть 172.16.3.0/24, то в плане адресации уже не было бы места для большей подсети.

Вторая возможность, вполне вероятно, сработает хорошо. Инженер находит неиспользуемые IP-адреса в той же сети IP и выбирает новую подсеть. Но придется изменить все существующие IP-адреса. Это относительно простой процесс, если большинство хостов использует протокол DHCP, но потенциально трудоемкий, если большинство хостов использует статические IP-адреса.

Третья возможность подразумевает использование *вторичной IP-адресации* маршрутизатора Cisco. Вторичная адресация позволяет использовать несколько сетей или подсетей на том же канале связи. (Это фактически нарушает правила создания подсетей, обсуждавшиеся ранее в этой книге, но на деле работает.) При использовании нескольких подсетей на равноправном уровне 2 широковещательного домена увеличивается количество доступных IP-адресов.

Концепции вторичной адресации приведены на рис. 16.14. Хосты A и B находятся в той же сети LAN, а фактически в той же сети VLAN. Это обеспечивает маршрутизатор R1. Никакого магистрального соединения также не нужно. Фактически, если игнорировать номера, хосты A, B и маршрутизатор R1 обычно являются частью той же подсети.

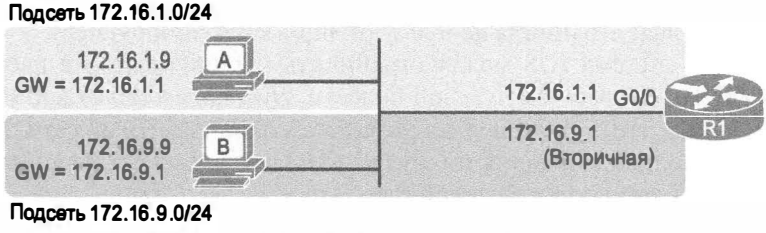


Рис. 16.14. Сеть TCP/IP со вторичными адресами

Вторичная адресация позволяет одним хостам иметь адреса в одной подсети IP, а другим во второй подсети IP, но маршрутизатор будет иметь адреса в обеих сетях. Обе подсети IP находятся на равноправном уровне 2 широковещательного домена (VLAN).

В результате у маршрутизатора будут подключенные маршруты для обеих подсетей, поэтому он сможет перенаправлять пакеты на обе подсети и даже между ними.

Пример 16.7 демонстрирует конфигурацию маршрутизатора R1, соответствующую схеме на рис. 16.14. У второй команды `ip address` должно быть ключевое слово `secondary`, реализующее вторичную адресацию и указывающее маршрутизатору добавлять этот IP-адрес как дополнительный. Без этого ключевого слова маршрутизатор заменил бы IP-адрес новым.

Пример 16.7. Настройка вторичной IP-адресации и команда `show ip route`

```
! Отрывок из файла running-config...
interface gigabitethernet 0/0
  ip address 172.16.9.1 255.255.255.0 secondary
  Ip Address 172.16.1.1 255.255.255.0

R1# show ip route connected
! Строки опущены для краткости
      172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
C     172.16.9.0/24 is directly connected, GigabitEthernet0/0
L     172.16.9.1/32 is directly connected, GigabitEthernet0/0
```

У вторичной адресации есть один недостаток: передача трафика между хостами в той же сети VLAN, но в разных подсетях требует прохождения маршрутизатора. Например, когда хост A в подсети 172.16.1.0 (см. рис. 16.14) посылает пакет на хост B в подсети 172.16.9.0, то хост A должен послать пакет на свой стандартный шлюз. В результате передающий хост отправляет пакет на маршрутизатор, который затем передает пакет на хост B, находящийся в другой подсети IP, но в сети VLAN на равноправном уровне 2.

Поддержка подключенных маршрутов для подсети нуль

В этом разделе обсуждается еще одно средство, позволяющее сетевым инженерам соединять маршрутизаторы с локальными подсетями так, чтобы маршрутизатор мог обмениваться пакетами с этими подсетями (настройка IP-адреса на физических интерфейсах с использованием схемы ROAS, коммутации третьего уровня и вторичной адресации). В данном разделе описано средство маршрутизатора, позволявшее на протяжении довольно долгого времени преодолевать некоторые из ранних проблем протокола IPv4. Сейчас его иногда используют лишь как крайнюю меру.

Операционная система IOS может ограничить маршрутизатор в настройке команды `ip address` с адресом в нулевой подсети. *Нулевая подсеть* (zero subnet) (или *подсеть нуль* (subnet zero)) — это первая подсеть в любой классовой сети, в ее номере часть подсети заполнена двоичными нулями. В десятичном виде номер нулевой подсети совпадает с номером классовой сети.

ВНИМАНИЕ!

Более подробная информация о концепции нулевых подсетей приведена в главе 19.

Операционная система IOS позволяет сетевому инженеру запретить или разрешить маршрутизатору использовать адреса в нулевой подсети. Причина в том, что

некоторые устаревшие протоколы маршрутизации IP не поддерживают использование нулевой подсети. Команда `ip subnet-zero` позволяет операционной системе IOS использовать нулевую подсеть без ограничений, а команда `no ip subnet-zero` заставит маршрутизатор отклонить любую команду `ip address`, использующую комбинацию адрес/маска для нулевой подсети. В большинстве современных версий IOS использование нулевой подсети разрешено.

Пример 16.8 демонстрирует, что с первоначальными настройками маршрутизатор принимает подкоманду интерфейса `ip address 10.0.0.1 255.255.255.0`, а после команды `no ip subnet-zero` отбрасывает ее. Обратите внимание, что сообщение об ошибке не упоминает нулевую подсеть, а просто заявляет, что это "плохая маска".

Пример 16.8. Воздействие команды `[no] ip subnet-zero` на локальный маршрутизатор

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface g0/1
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# no ip address
R1(config-if)# exit
R1(config)# no ip subnet-zero
R1(config)# interface g0/1
R1(config-if)# ip address 10.0.0.1 255.255.255.0
Bad mask /24 for address 10.0.0.1
```

Хотя команда `no ip subnet-zero` влияет на команды `ip address` и `ip route` локального маршрутизатора, определяющие статические маршруты, она не затрагивает маршруты локального маршрутизатора, изученные по протоколу маршрутизации. Например, на маршрутизаторе R1 могла бы быть введена команда `no ip subnet-zero`, но, используя протокол маршрутизации, он вполне может изучить маршрут к нулевой подсети.

Настройка статических маршрутов

Все маршрутизаторы добавляют в таблицы маршрутизации подключенные маршруты, как обсуждалось в предыдущем разделе. Далее, в большинстве сетей используются протоколы динамической маршрутизации, позволяющие каждому маршрутизатору изучать остальную часть маршрутов в объединенной сети. Сети используют статические маршруты (добавляемые в таблицу маршрутизации вручную) много реже, чем динамические. Но иногда статические маршруты могут быть очень полезны, поэтому имеет смысл изучить этот инструмент. Данный раздел посвящен обсуждению статических маршрутов.

Конфигурация статического маршрута

Операционная система IOS позволяет задавать отдельные статические маршруты при помощи глобальной команды конфигурации `ip route`. Каждая команда `ip route` определяет получателя, задаваемого, как обычно, идентификатором подсети и маской. Команда включает также инструкции перенаправления, как правило, исходящий интерфейс или IP-адрес следующего транзитного маршрутизатора. Опера-

ционная система IOS получает эту информацию и добавляет маршрут в таблицу маршрутизации IP.

В качестве примера на рис. 16.15 приведена небольшая сеть IP. Фактически это фрагмент сети на рис. 16.3, но без некоторых малозначительных деталей. Здесь представлены подробности статического маршрута к подсети 172.16.2.0/24 на маршрутизаторе R1 (на рисунке крайний справа). Для создания этого статического маршрута на маршрутизаторе R1 необходимо задать идентификатор подсети и маску, а также исходящий интерфейс (S0/0/0) маршрутизатора R1, или IP-адрес следующего транзитного маршрутизатора R2 (172.16.4.2).

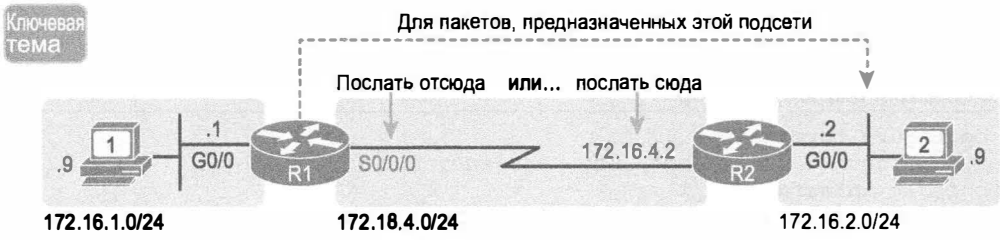


Рис. 16.15. Концепция настройки статического маршрута

Пример 16.9 демонстрирует настройку нескольких типичных статических маршрутов. В частности, маршрутов на маршрутизаторе R1 для двух подсетей справа на рис. 16.16.

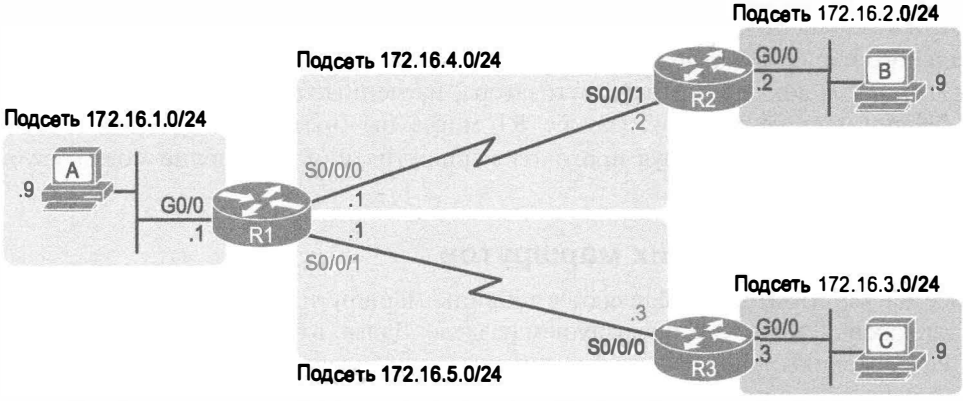


Рис. 16.16. Сеть, используемая в примерах настройки статического маршрута

Пример 16.9. Статические маршруты, добавленные на маршрутизатор R1

```
ip route 172.16.2.0 255.255.255.0 172.16.4.2
ip route 172.16.3.0 255.255.255.0 S0/0/1
```

Два примера демонстрируют два различных стиля команды `ip route`. Первая команда относится к подсети 172.16.2.0 и маске 255.255.255.0, расположенной в сети LAN около маршрутизатора R2. В качестве следующего транзитного маршрутизатора эта команда указывает IP-адрес 172.16.4.2 маршрутизатора R2. Таким образом, этот маршрут утверждает: чтобы передать пакеты в подсеть маршрутизатора R2, пошлите их на маршрутизатор R2.

Логика второго маршрута точно такая же, но вместо IP-адреса следующего маршрутизатора указан исходящий интерфейс локального маршрутизатора. Этот маршрут утверждает: чтобы передать пакеты в подсеть маршрутизатора R3, пошлите их на локальный интерфейс S0/0/1 (который подключен к маршрутизатору R3).

Маршруты, созданные этими двумя командами `ip route`, в таблице маршрутизации IP выглядят немного по-разному. Оба являются статическими маршрутами. Однако маршрут с конфигурацией исходящего интерфейса является также и подключенным маршрутом, как свидетельствует вывод команды `show ip route`.

Команда `show ip route static` выводит в примере 16.10 оба эти маршрута. Данная команда выводит подробности только статических маршрутов, а об остальных маршрутах IPv4 выводит лишь статистические данные. Пример демонстрируют две строки вывода для двух статических маршрутов, настроенных в примере 16.9, но в статистике перенаправления есть маршруты для десяти подсетей.

Пример 16.10. Статические маршруты, добавленные маршрутизатором R1

```
R1# show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
! Строки опущены для краткости
Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 10 subnets, 2 masks
S     172.16.2.0/24 [1/0] via 172.16.4.2
S     172.16.3.0/24 is directly connected, Serial0/0/1
```

В зависимости от того, работает ли исходящий интерфейс, операционная система IOS добавляет и удаляет эти статические маршруты динамически. В данном случае, например, при отказе интерфейса S0/0/1 маршрутизатора R1 статический маршрут к подсети 172.16.3.0/24 удаляется из таблицы маршрутизации IPv4. Впоследствии, когда интерфейс заработает снова, IOS добавит маршрут в таблицу маршрутизации снова. Обратите также внимание на то, что команда `ip route` поддерживает ключевое слово `permanent`, требующее оставить статический маршрут в таблице маршрутизации, даже если соответствующий интерфейс отключен.

И наконец, при использовании статических маршрутов, но без применения протоколов динамической маршрутизации, на всех маршрутизаторах должны были бы быть настроены статические маршруты. На настоящий момент в сети на рис. 16.16 компьютер A не был бы в состоянии получать пакеты от компьютера B, поскольку у маршрутизатора R2 нет маршрута для подсети компьютера A. Маршрутизатору R2 понадобятся статические маршруты и для других подсетей, как и маршрутизатору R3.

Статические стандартные маршруты

При попытке перенаправить пакет маршрутизатор может не найти маршрут к IP-адресу получателя пакета. Обычно в таком случае маршрутизатор просто отбрасывает пакет.

Маршрутизаторы могут быть настроены так, чтобы использовались либо статически заданные маршруты, либо динамически изученный стандартный маршрут. Стандартный маршрут соответствует всем пакетам, поэтому, если пакет не соответствует никакому другому более специфическому маршруту в таблице маршрутизации, маршрутизатор перенаправляет его на стандартный маршрут.

Вот классический пример использования статических стандартных маршрутов в корпоративной сети TCP/IP: у компании много дистанционных площадок с индивидуальными относительно медленными соединениями WAN. У каждой дистанционной площадки есть только один возможный физический маршрут передачи пакетов в остальную часть сети. Поэтому вместо протокола маршрутизации, рассылающего сообщения по каналам WAN и растрачивающего их полосу пропускания, каждый дистанционный маршрут мог бы использовать стандартный маршрут, посылающий весь трафик на центральную площадку, как показано на рис. 16.17.

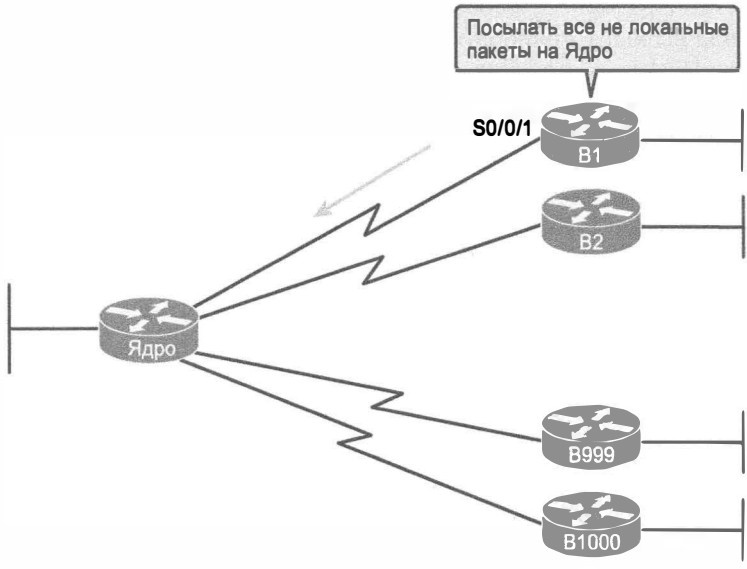


Рис. 16.17. Пример применения статических стандартных маршрутов: 1000 медленных дистанционных площадок

Операционная система IOS позволяет настроить статический стандартный маршрут за счет специальных значений (0.0.0.0 и 0.0.0.0) полей подсети и маски в команде `ip route`. Например, команда `ip route 0.0.0.0 0.0.0.0 S0/0/1` создает на маршрутизаторе B1 статический стандартный маршрут (соответствующий всем пакетам IP) и посылает эти пакеты на интерфейс S0/0/1.

Пример 16.11 демонстрирует статический стандартный маршрут для маршрутизатора R2, представленного на рис. 16.16. Ранее этот рисунок наряду с примером 16.9 представил маршрутизатор R1 со статическими маршрутами к двум подсетям, показанным справа. Пример 16.11 демонстрирует маршрутизатор R2 (справа), использующий статический стандартный маршрут для перенаправления пакетов назад в левую часть рисунка.

Пример 16.11. Добавление статического стандартного маршрута для маршрутизатора R2 (рис. 16.16)

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ip route 0.0.0.0 0.0.0.0 s0/0/1
```

```
R2(config)# ^Z
R2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1
       L2 - IS-IS level-2, ia - IS-IS inter area, * - candidate default,
       U - per-user static route, o - ODR,
       P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

S* 0.0.0.0/0 is directly connected, Serial0/0/1
       172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C      172.16.2.0/24 is directly connected, GigabitEthernet0/0
L      172.16.2.2/32 is directly connected, GigabitEthernet0/0
C      172.16.4.0/24 is directly connected, Serial0/0/1
L      172.16.4.2/32 is directly connected, Serial0/0/1
```

Вывод команды `show ip route` отображает несколько новых интересных фактов. В первую очередь он выводит маршрут с кодом "S", означающим *статический* (static), а также со звездочкой (`*`), означающей, что это *кандидат в стандартные маршруты*. Маршрутизатор может узнать о нескольких стандартных маршрутах, поэтому ему следует выбрать, какой использовать; звездочка означает, что это по крайней мере кандидат в стандартные маршруты. Выбранный стандартный маршрут указан выше, в разделе "`Gateway of last resort`", которым в данном случае является статически настроенный маршрут с исходящим интерфейсом S0/0/1.

Обзор

Резюме

- Маршрутизация IP (процесс перенаправления пакетов IP) обеспечивает доставку пакетов через все сети TCP/IP с устройства, создавшего пакет IP, на устройство его получателя. Другими словами, маршрутизация IP доставляет пакеты IP с хоста отправителя на хост получателя.
- Процесс маршрутизации начинается с хоста, создающего пакет IP. Сначала хост решает вопрос: не принадлежит ли IP-адрес получателя этого нового пакета локальной подсети? Для определения диапазона адресов в локальной подсети хост использует собственный IP-адрес и маску. На основании собственных выводов о диапазоне адресов локальной подсети хост действует следующим образом.

  Этап 1 Если получатель локальный, передача осуществляется непосредственно

  A. MAC-адрес хоста получателя определяется при помощи уже существующей записи таблицы протокола преобразования адресов (ARP) или сообщения ARP, позволяющего изучить эту информацию.

  B. Пакет IP инкапсулируется во фрейм канала связи с адресом канала связи хоста получателя

  Этап 2 Если получатель не является локальным, осуществляется передача на стандартный шлюз.

  A. MAC-адрес стандартного шлюза определяется при помощи уже существующей записи таблицы ARP или сообщения ARP, позволяющего изучить эту информацию.

  B. Пакет IP инкапсулируется во фрейм канала связи с адресом канала связи стандартного шлюза

- У маршрутизаторов немного больше работы при маршрутизации по сравнению с хостами. В то время как логика хоста начинается с пакета IP, находящегося в памяти, маршрутизатору, прежде чем дойти до того положения, необходимо проделать некоторую работу. Ниже приведены пять этапов логики маршрутизации, причем на первых двух этапах осуществляется только получение фрейма и извлечение пакета IP перед принятием решения об адресе получателя пакета на этапе 3.

  Этап 1 Для каждого полученного фрейма канала связи принимается решение, обрабатывать его или нет. Обрабатывается фрейм так:

  A. Проверка фрейма на ошибки (по полю контрольной суммы фрейма (FCS) в концевике канала связи).

  B. Адрес канала связи получателя фрейма — это адрес маршрутизатора (или соответствующий многоадресатный или широковещательный адрес)

  Этап 2 Перед принятием решения об обработке фрейма на этапе 1 он извлекается из фрейма канала связи

Этап 3 Принимается решение о маршрутизации. Для этого по IP-адресу получателя пакета осуществляется поиск соответствующего элемента таблицы маршрутизации, содержащего маршрут к получателю. Этот маршрут идентифицирует исходящий интерфейс маршрутизатора, а возможно, и следующий транзитный маршрутизатор

Этап 4 Помещает (инкапсулирует) пакет во фрейм канала связи, соответствующего исходящему интерфейсу. По мере необходимости для поиска MAC-адреса следующего устройства используется протокол ARP

Этап 5 Фрейм передается на исходящий интерфейс, указанный в соответствующем маршруте IP

- Отбросив некоторые детали, этапы этого процесса можно коротко пересказать так: маршрутизатор получает фрейм, извлекает из него пакет, решает, куда его перенаправить, помещает пакет в другой фрейм и посылает его.
- После того как маршрутизатор сможет перенаправлять пакеты IP через один или несколько интерфейсов, ему понадобятся маршруты. Маршрутизаторы могут добавлять маршруты в свои таблицы маршрутизации тремя способами.
  - *Подключенные маршруты.* Добавляются подкомандой интерфейса `ip address` на локальном маршрутизаторе.
  - *Статические маршруты.* Добавляются глобальной командой конфигурации `ip route` на локальном маршрутизаторе.
  - *Протоколы маршрутизации.* Дополнительная функция настройки на всех маршрутизаторах, обеспечивающая динамический обмен данными о сети между маршрутизаторами, позволяющий им изучить все маршруты.
- Операционная система IOS может ограничить маршрутизатор в настройке команды `ip address` с адресом в нулевой подсети. Нулевая подсеть (или подсеть нуль) — это первая подсеть в любой классовой сети, в ее номере часть подсети заполнена двоичными нулями. В десятичном виде номер нулевой подсети совпадает с номером классовой сети.
- При попытке перенаправить пакет маршрутизатор может не найти маршрут к IP-адресу получателя пакета. Обычно в таком случае маршрутизатор просто отбрасывает пакет.
- Маршрутизаторы могут быть настроены так, чтобы использовались либо статически заданные маршруты, либо динамически изученный стандартный маршрут. Стандартный маршрут соответствует всем пакетам, поэтому, если пакет не соответствует никакому другому более специфическому маршруту в таблице маршрутизации, маршрутизатор перенаправляет его на стандартный маршрут.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. На компьютере открыто приглашение к вводу и введена команда `ipconfig`, показавшая IP-адрес 192.168.4.77 и маску 255.255.255.224 компьютера. Затем пользователь осуществляет проверку командой `ping 192.168.4.117`. Какой из следующих ответов вероятней всего справедлив?

А) Компьютер посылает пакеты на хост с адресом 192.168.4.117 непосредственно.

Б) Компьютер посылает пакеты на свой стандартный шлюз.

В) Компьютер посылает запрос DNS на адрес 192.168.4.117.

Г) Компьютер посылает запрос ARP на поиск MAC-адреса сервера DHCP.

2. Маршрутизатор R1 содержит маршруты в таблице маршрутизации. В каком из следующих ответов указано то, что маршрутизатор сравнивает с адресом получателя пакета? (Выберите два ответа).

А) Маска.

Б) Следующий транзитный маршрутизатор.

В) Идентификатор подсети.

Г) Исходящий интерфейс.

3. У маршрутизатора 1 есть интерфейс FastEthernet 0/0 с IP-адресом 10.1.1.1. Интерфейс подключен к коммутатору. В последующем проводится модернизация этого подключения для использования магистрального соединения 802.1Q. Какие из указанных ниже команд могут быть частью допустимой конфигурации интерфейса Fa0/0 для маршрутизатора 1? (Выберите два ответа.)

А) `interface fastethernet 0/0.4.`

Б) `dot1q enable.`

В) `dot1q enable 4.`

Г) `trunking enable.`

Д) `trunking enable 4.`

Е) `encapsulation dot1q 4.`

4. Конфигурация маршрутизатора настроена с применением глобальной команды конфигурации `no ip subnet-zero`. Какую из следующих команд интерфейса нельзя ввести в конфигурацию маршрутизатора?

А) `ip address 10.1.1.1 255.255.255.0.`

Б) `ip address 10.0.0.129 255.255.255.128.`

В) `ip address 10.1.2.2 255.254.0.0.`

Г) `ip address 10.0.0.5 255.255.255.252.`

5. Коммутатор уровня 3 был настроен на перенаправление пакетов IP между сетями VLAN 1, 2 и 3, подключенными к подсетям 172.20.1.0/25, 172.20.2.0/25 и 172.20.3.0/25 соответственно. На коммутаторе уровня 3 команда `show ip route` выводит подключенные маршруты. Какой из следующих ответов перечисляет информацию, присущую по крайней мере одному из маршрутов?

А) Интерфейс GigabitEthernet 0/0.3.

Б) Следующий транзитный маршрутизатор 172.20.4.1.

В) Интерфейс VLAN 2.

Г) Маска 255.255.255.0.

6. На маршрутизаторе R1 настроен статический маршрут IPv4. Какой из следующих информационных элементов не должен быть параметром команды конфигурации, создающей статический маршрут IPv4?

А) Идентификатор подсети получателя.
Б) IP-адрес следующего транзитного маршрутизатора.
В) Интерфейс следующего транзитного маршрутизатора.
Г) Маска подсети.

7. Какая из следующих команд корректно настраивает статический маршрут?

А) `ip route 10.1.3.0 255.255.255.0 10.1.130.253`
Б) `ip route 10.1.3.0 serial 0`
В) `ip route 10.1.3.0 /24 10.1.130.253`
Г) `ip route 10.1.3.0 /24 serial 0`

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 16.2.

Таблица 16.2. Ключевые темы главы 16

Элемент	Описание	Страница
Список	Этапы перенаправления пакетов IP хостом	469
Список	Этапы перенаправления пакетов IP маршрутизатором	469
Рис. 16.2	Пять этапов маршрутизации, осуществляемых маршрутизатором	470
Рис. 16.7	3-й этап маршрутизации: таблица маршрутизации IP имеет части соответствия и перенаправления	474
Список	Три источника маршрутов IP для маршрутизаторов	477
Список	Правила создания маршрутизатором подключенного маршрута	478
Список	Три возможности соединения маршрутизаторов с каждой сетью VLAN	480
Рис. 16.12	Субинтерфейсы на маршрутизаторе B1	481
Список	Этапы настройки магистрального соединения 802.1Q	481
Список	Последовательность настройки собственной сети VLAN 802.1	483
Рис. 16.13	Маршрутизация на интерфейсах VLAN коммутатора уровня 3	485
Список	Настройка коммутации третьего уровня	485
Рис. 16.15	Концепция настройки статического маршрута	490

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

стандартный шлюз (default gateway), стандартный маршрутизатор (default router), таблица ARP (ARP table), таблица маршрутизации (routing table), следующий транзитный маршрутизатор (next-hop router), исходящий интерфейс (outgoing interface), субинтерфейс (subinterface), интерфейс VLAN (VLAN interface), коммутатор третьего уровня (Layer 3 switch), экспресс-передача Cisco (Cisco Express Forwarding — CEF), подключенный маршрут (connected route), статический маршрут (static route), стандартный маршрут (default route), нулевая подсеть (zero subnet)

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 16.3 приведен список команд конфигурации, а в табл. 16.4 приведены пользовательские команды главы. Команды стоит запомнить, чтобы лучше понять материал главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 16.3. Команды конфигурации главы 16

Команда	Описание
`ip address` *ip-адрес маска* `[secondary]`	Подкоманда интерфейса, присваивающая интерфейсу IP-адрес, а также способная сделать адрес вторичным
`interface` *тип номер.подинт*	Глобальная команда создания субинтерфейса и перехода в режим конфигурации данного субинтерфейса
`encapsulation dot1q` *идентификатор_vlan* `[native]`	Подкоманда субинтерфейса, указывающая маршрутизатору использовать магистральное соединение 802.1Q для определенной сети VLAN с ключевым словом `native`, чтобы избежать инкапсуляции в магистральный заголовок
`encapsulation isl` *идентификатор_vlan*	Подкоманда субинтерфейса, указывающая маршрутизатору использовать магистраль ISL для определенной сети VLAN
`sdm prefer lanbase-routing`	Команда коммутатора Cisco, разрешающая маршрутизацию IP
`[no] ip routing`	Глобальная команда, разрешающая (`ip routing`) и запрещающая (`no ip routing`) маршрутизацию пакетов IPv4 на маршрутизаторе или коммутаторе уровня 3
`interface vlan` *идентификатор_vlan*	Глобальная команда на коммутаторе уровня 3 для создания интерфейса VLAN и перехода в режим конфигурации для данного интерфейса VLAN
`[no] ip subnet-zero`	Глобальная команда, разрешающая (`ip subnet-zero`) и запрещающая (`no ip subnet-zero`) настройку IP-адреса интерфейса в нулевой подсети
`ip route` *префикс маска* {*ip-адрес* | *тип_интерфейса номер_интерфейса*} [*дистанция*] `[permanent]`	Глобальная команда конфигурации, создающая статический маршрут
`ip default-network` *номер_сети*	Глобальная команда, создающая стандартный маршрут на основании маршрута к классовой сети, указанной в команде

Таблица 16.4. Пользовательские команды главы 16

Команда	Описание
`show ip route`	Выводит всю таблицу маршрутизации маршрутизатора
`show ip route [connected | static | ospf]`	Выводит подсети таблицы маршрутизации IP
`show ip route` *ip-адрес*	Выводит подробную информацию о маршруте, соответствующем указанному IP-адресу
`show vlans`	Выводит конфигурацию и статистику магистральных каналов VLAN, настроенных на маршрутизаторах

Ответы на контрольные вопросы:

1 Б. **2** A и В. **3** A и Е. **4** В. **5** В. **6** В. **7** А.

Глава 17

Самообучение маршрутов IPv4 с использованием OSPFv2

Открытый протокол поиска первого кратчайшего маршрута (Open Shortest Path First — OSPF) может быть использован любым маршрутизатором для изучения маршрутов ко всем подсетям в корпоративной сети IPv4. Фактически при относительно простой конструкции OSPF маршрутизаторы могли бы использовать ту же конфигурацию OSPF с двумя командами: `router ospf 1` и `network 0.0.0.0 255.255.255.255 area 0`. Если единственной задачей является правильная работа протокола OSPF, откажитесь от желания понять его, можете пропустить эту главу, настраивать все маршрутизаторы и закончить на этом. (Соблазнительно, не так ли?)

Конечно, и для задач реальных сетей, и для экзаменов CCENT и CCNA следует понимать концепции и знать параметры настройки. Весь этот процесс рассматривается в данной главе. Она начинается со сравнения разных протоколов маршрутизации и связанных с ними концепций. Продолжается глава изложением теории протоколов маршрутизации по состоянию канала, поскольку протокол OSPF использует принципы состояния канала. В последнем разделе демонстрируется настройка основных параметров протокола OSPF и соответствующие команды `show`.

Обратите внимание, что версия 2 протокола OSPF (OSPFv2) является общепринятой версией для протокола IPv4, в то время как версия 3 протокола OSPF (OSPFv3) была специально разработана для поддержки протокола IPv6. В данной главе обсуждаются протоколы маршрутизации только для протокола IPv4. Таким образом, все упоминания протокола OSPF относятся фактически к протоколу OSPFv2. Более подробная информация о протоколе OSPF для протокола IPv6 приведена в главе 29.

В этой главе рассматриваются следующие экзаменационные темы

Технологии маршрутизации IP

- Настройка и проверка применения CLI для установки базовой конфигурации маршрутизатора.
- Команды Cisco IOS для базовой настройки маршрутизатора.
- Проверка конфигурации маршрутизатора и сетевого подключения.
- Команды Cisco IOS для просмотра базовой информации маршрутизатора и сетевого подключения.
- Различия методов маршрутизации и протоколов маршрутизации:
  - Статика или динамика.
  - Состояние канала или вектор расстояния.
  - Пассивные интерфейсы.
- Настройка и проверка OSPF (единая область).
  - Преимущество единой области.
  - Настройка OSPF v2.
  - Идентификатор маршрутизатора.
  - Пассивный интерфейс.

Основные темы

Сравнение средств протокола динамической маршрутизации

Маршрутизаторы добавляют маршруты IP в свои таблицы маршрутизации тремя способами: как подключенные и статические маршруты и изучает их при помощи протоколов динамической маршрутизации. Прежде чем углубляться в обсуждение темы, имеет смысл определить несколько связанных с ними терминов и устранить заблуждения по поводу терминов *протокол маршрутизации* (routing protocol), *маршрутизируемый протокол* (routed protocol и routable protocol). Лежащие в основе этих терминов концепции вовсе не трудны, просто сами термины настолько похожи, что во многих документах их не различают и путают. Обобщенно эти термины определяют следующим образом.

- *Протокол маршрутизации* (routing protocol). Набор сообщений, правил и алгоритмов, используемых маршрутизаторами для общей задачи изучения маршрутов. Этот процесс подразумевает обмен и анализ информации о маршрутизации. Каждый маршрутизатор выбирает наилучший маршрут к каждой подсети (выбор пути), а выбранные оптимальные маршруты помещает в свою таблицу маршрутизации IP. К таким протоколам относятся RIP, EIGRP, OSPF и BGP.
- *Маршрутизируемый протокол* (routed protocol и routable protocol). Оба термина описывают протокол, определяющий структуру пакета и логику адресации, позволяющие маршрутизаторам перенаправлять (или маршрутизировать) пакеты. Маршрутизаторы перенаправляют пакеты в соответствии с маршрутизируемыми протоколами. К таким протоколам относятся IP версии 4 (IPv4) и версии 6 (IPv6).

ВНИМАНИЕ!

Термин *выбор пути* (path selection) иногда применяют к части задач протокола маршрутизации, подразумевающей выбор протоколом маршрутизации наилучшего маршрута.

Даже при том, что протоколы маршрутизации (такие, как OSPF) отличаются от маршрутизируемых протоколов (таких, как IP), взаимодействуют они очень тесно. Процесс маршрутизации обеспечивает перенаправление пакетов IP, но если у маршрутизатора нет в таблице маршрутизации IP маршрута, соответствующего адресу получателя пакета, то маршрутизатор отбрасывает пакет. Чтобы маршрутизаторы могли изучить все возможные маршруты и добавить их к таблице маршрутизации, они нуждаются в протоколах маршрутизации. Это позволит процессу маршрутизации перенаправлять пакеты, используя такие протоколы, как IP.

Функции протокола маршрутизации

Программное обеспечение Cisco IOS поддерживает несколько протоколов маршрутизации IP, осуществляющих те же общие функции.

Ключевая тема

Список основных функций протокола маршрутизации

1. Изучать информацию о маршрутах к подсетям IP от других соседних маршрутизаторов.
2. Анонсировать информацию о маршрутах к подсетям IP другим соседним маршрутизаторам.
3. Если к подсети возможно несколько маршрутов, то наилучший выбирается на основании метрик.
4. Если топология сети изменяется (например, откажет канал связи), то некоторые маршруты будут неверны и придется выбирать новый оптимальный маршрут. (Этот процесс носит название *конвергенция* (convergence).)

ВНИМАНИЕ!

Соседний маршрутизатор подключен к тому же каналу связи, что и другой маршрутизатор, например, к тому же каналу связи WAN или к тому же каналу Ethernet LAN.

На рис. 17.1 приведен пример трех из четырех приведенных выше функций. Маршрутизаторы R1 и R3 узнают о маршруте к подсети 172.16.3.0/24 от маршрутизатора R2 (функция 1). После того как маршрутизатор R3 узнает о маршруте к подсети 172.16.3.0/24 от маршрутизатора R2, он анонсирует этот маршрут маршрутизатору R1 (функция 2). Затем маршрутизатор R1 принимает решение о двух изученных им маршрутах к подсети 172.16.3.0/24: один с метрикой 1 от маршрутизатора R2 и один с метрикой 2 от маршрутизатора R3. Маршрутизатор R1 выбирает маршрут с меньшей метрикой, т.е. через маршрутизатор R2 (функция 3).

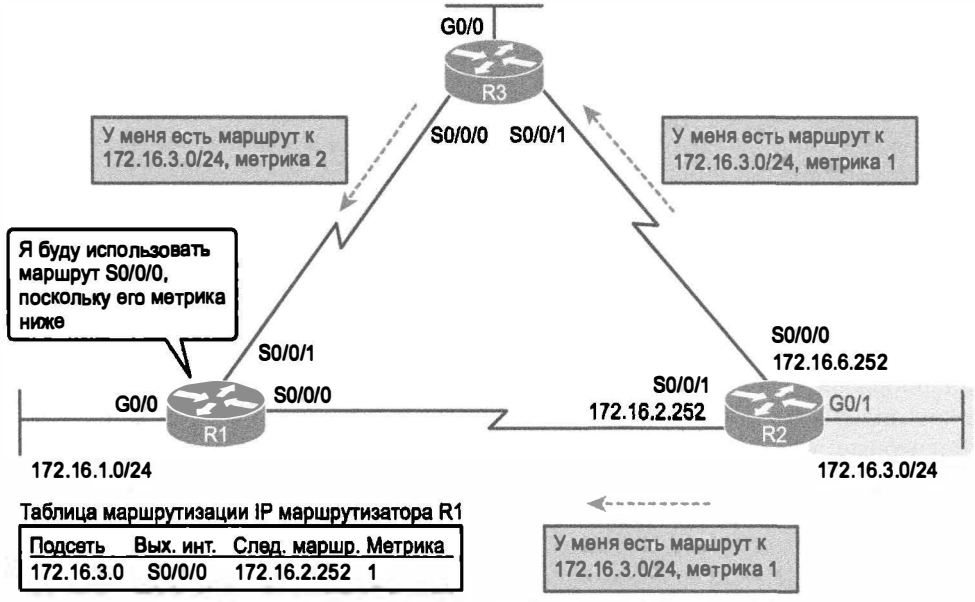


Рис. 17.1. Три из четырех базовых функций протоколов маршрутизации

Конвергенция — четвертая упомянутая здесь функция протокола маршрутизации. Термин *конвергенция* (convergence) относится к процессу, происходящему при изменении топологии сети, т.е. когда отказывает маршрутизатор или канал связи (либо когда они восстанавливают работу). Когда нечто изменяется, могут измениться и наилучшие доступные в сети маршруты. Конвергенция — это процесс, в ходе которого все маршрутизаторы осознают, что произошло некое изменение, анонсируют информацию об изменениях всем остальным маршрутизаторам, а они выбирают для каждой подсети новые оптимальные маршруты. Способность к быстрой конвергенции (без циклов) является одним из важнейших критериев при выборе протокола маршрутизации IP.

Конвергенция на рис. 17.1 могла бы иметь место при повреждении канала связи между маршрутизаторами R1 и R2. В этом случае маршрутизатор R1 прекратит использовать свой прежний маршрут для подсети 172.16.3.0/24 (непосредственно через маршрутизатор R2) и начнет посылать пакеты через маршрутизатор R3.

Внутренние и внешние протоколы маршрутизации

Протоколы маршрутизации IP относятся к одной из двух главных категорий: *протоколы маршрутизации внутреннего шлюза* (Interior Gateway Protocol — IGP) и *протоколы маршрутизации внешнего шлюза* (Exterior Gateway Protocol — EGP). Как они определяются, описано ниже.

Ключевая тема

Определения протоколов IGP и EGP

- *IGP*. Протокол маршрутизации, предназначенный для использования в одиночной *автономной системе* (Autonomous System — AS).
- *EGP*. Протокол маршрутизации, предназначенный для совместного использования разными автономными системами.

ВНИМАНИЕ!

Термины *IGP* и *EGP* включают слово "шлюз" потому, что маршрутизаторы раньше называли шлюзами.

В этих определениях использован еще один новый термин: *автономная система* (Autonomous System — AS). Автономная система — это сеть под административным контролем одной организации. Например, сеть, созданная и оплаченная некой компанией, будет вероятней всего системой AS, школьная сеть также, вероятно, является автономной системой. Другими примерами являются разнообразные государственные или национальные учреждения, вполне способные создавать собственные сети. Каждый провайдер ISP также обычно является отдельной автономной системой.

Некоторые протоколы маршрутизации лучше работают в автономной системе (согласно проекту), поэтому их относят к протоколам IGP. И наоборот, протоколы маршрутизации, предназначенные для обмена маршрутами между маршрутизаторами в разных автономных системах, относят к протоколам EGP. Ныне *протокол граничного шлюза* (Border Gateway Protocol — BGP) является единственным используемым протоколом EGP.

Каждой системе AS может быть присвоен *номер AS* (AS number — ASN). Правом присвоения номеров ASN, как и открытых IP-адресов, обладает *центр по присвоению адресов Интернета* (Internet Assigned Numbers Authority — IANA `www.iana.org`). Центр делегирует свои полномочия другим организациям по всему миру, как правило, тем же организациям, которые присваивают открытые IP-адреса. Например, в Северной Америке этим занимается American Registry for Internet Numbers (ARIN) `www.arin.net`.

На рис. 17.2 приведено некое уменьшенное представление всемирного Интернета. На рисунке представлены два предприятия и три провайдера услуг Интернета, использующих протоколы IGP (OSPF и EIGRP) в собственных сетях и протоколы BGP между системами, обозначенными как ASN.

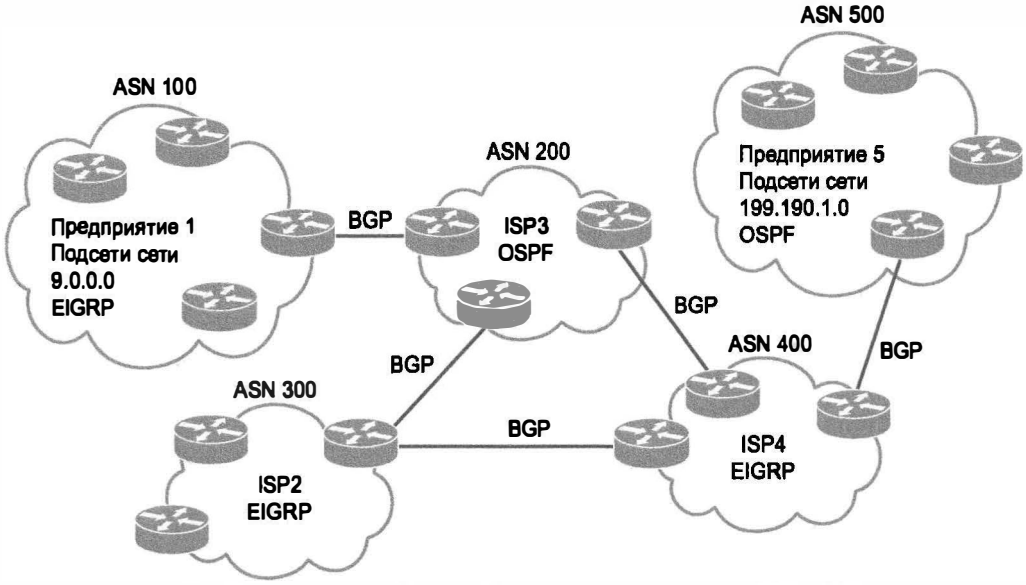


Рис. 17.2. Сравнение областей применения протоколов IGP и EGP

Сравнение протоколов IGP

У предприятий есть несколько вариантов при выборе протокола IGP для своей корпоративной сети, но большинство компаний ныне использует протокол OSPF или EIGRP. В этой книге обсуждается протокол OSPF, а протокол EIGRP рассматривается в книге по ICND2. Хотя эти книги сравнивают и противопоставляют эти протоколы IGP, в данном разделе обсуждаются сначала некоторые из основных задач любого протокола IGP, включая протоколы OSPF, EIGRP и многие другие протоколы маршрутизации IPv4.

Алгоритмы протокола маршрутизации IGP

Базовые алгоритмы протокола маршрутизации определяют то, как они решают свою задачу. Термин *алгоритм протокола маршрутизации* (routing protocol algorithm) относится к логике, используемой различными протоколами маршрутизации для изучения всех возможных маршрутов и выбора наилучшего для каж-

дой подсети, а также реакции на изменения в объединенной сети (конвергенции). Протоколы маршрутизации IGP используют три основных типа алгоритмов протокола маршрутизации.

Ключевая тема **Три типа алгоритмов протокола маршрутизации IGP**

- Дистанционно-векторный (или алгоритм Беллмана–Форда, по имени его создателей).
- Улучшенный дистанционно-векторный (или "сбалансированный гибридный").
- Состояния канала.

Исторически первыми были изобретены дистанционно-векторные протоколы в начале 1980-х годов. *Протокол маршрутной информации* (Routing Information Protocol — RIP) был первым популярным дистанционно-векторным протоколом, используемым протоколом IP, а немного позже появился собственный протокол компании Cisco — *протокол маршрутизации внутреннего шлюза* (Interior Gateway Routing Protocol — IGRP).

К началу 1990-х дистанционно-векторные протоколы проявили медленную конвергенцию и вероятность циклических маршрутов. Это потребовало разработки новых протоколов маршрутизации, использующих альтернативные алгоритмы. Основные проблемы решили протоколы состояния канала, в частности *открытый протокол поиска первого кратчайшего маршрута* (Open Shortest Path First — OSPF) и протокол *IS-IS* (Integrated Intermediate System to Intermediate System). Но у всего своя цена: они потребовали дополнительной мощности процессора и памяти маршрутизатора, а также более тщательного планирования от сетевых инженеров.

Почти одновременно с введением протокола OSPF компания Cisco выпустила собственный *расширенный протокол маршрутизации внутреннего шлюза* (Enhanced Interior Gateway Routing Protocol — EIGRP), использовавший некоторые из средств прежнего протокола маршрутизации внутреннего шлюза. Протокол EIGRP решал те же задачи, что и протоколы маршрутизации по состоянию канала, но при меньшем планировании реализуемой сети. Со временем протокол EIGRP был классифицирован как уникальный тип протокола маршрутизации. Но поскольку он больше использовал дистанционно-векторные средства, чем средства состояния канала, его обычно относят скорее к улучшенным дистанционно-векторным протоколам.

Метрики

Протоколы маршрутизации выбирают наилучший маршрут к подсети на основании самой низкой метрики маршрута. Протокол RIP, например, использует счетчик количества маршрутизаторов (транзитных участков) между текущим маршрутизатором и подсетью назначения. Протокол OSPF подсчитывает цену каждого интерфейса в сквозном маршруте, а также цену на основании ширины полосы пропускания канала связи. Список важнейших протоколов маршрутизации IP для экзаменов CCNA и некоторые подробности о них приведены в табл. 17.1.

Таблица 17.1. Сравнение метрик IGP

Ключевая тема

IGP	Метрика	Описание
RIP-2	Счетчик транзитных переходов	Количество маршрутизаторов (транзитных участков) между маршрутизатором и подсетью назначения
OSPF	Цена	Сумма стоимостей всех интерфейсов для всех каналов связи на маршруте со стоимостью, полученной на основании ширины полосы пропускания интерфейса
EIGRP	Соотношение ширины полосы пропускания и задержки	Вычисляется на основании самого медленного канала связи маршрута, а кумулятивная задержка связана с каждым интерфейсом на маршруте

Хотя современные экзамены CCENT и CCNA игнорируют протокол RIP, краткое сравнение его метрик с метриками протокола EIGRP позволяет лучше понять, почему протоколы OSPF и EIGRP превзошли протокол RIP. На рис. 17.3 приведен пример, где маршрутизатор B имеет два возможных маршрута к подсети 10.1.1.0 с левой стороны: более короткий маршрут по медленному каналу связи (64 Кбайт/с) и более длинный маршрут по высокоскоростным каналам связи (T1).

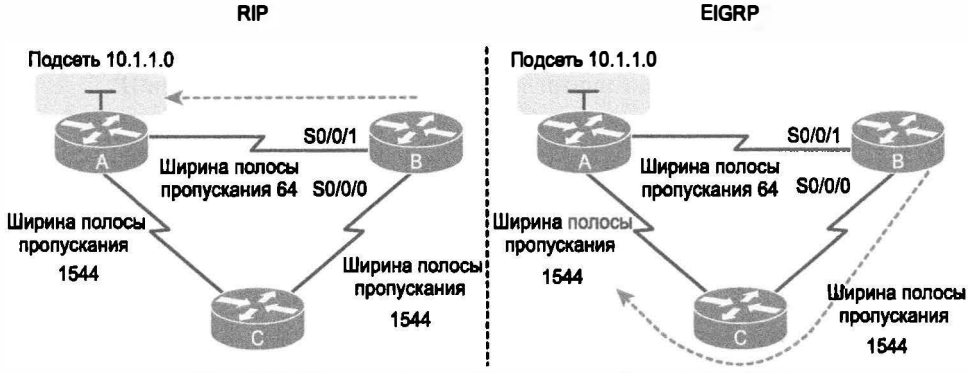


Рис. 17.3. Сравненные метрик протоколов RIP и EIGRP

Слева на рисунке приведен результат работы протокола RIP. Используя счетчик переходов, маршрутизатор B выяснил и пришел к выводу, что маршрут к маршрутизатору A через интерфейс S0/0/1 короче (1 транзитный участок). Маршрутизатор B находит также маршрут из двух переходов через маршрутизатор C (интерфейс S0/0/0), но выбирает маршрут с наименьшим значением счетчика переходов, хотя он и проходит через очень медленный канал связи.

Справа на рисунке представлен, возможно, наилучший выбор, сделанный протоколом EIGRP на основании его метрик.

Чтобы протокол EIGRP делал правильный выбор, инженер должен правильно настроить ширину полосы пропускания интерфейса, чтобы она соответствовала фактической скорости канала, позволяя протоколу EIGRP выбрать более быстрый маршрут. (Подкоманда интерфейса `bandwidth` не изменяет фактическую физическую скорость интерфейса. Она указывает операционной системе IOS, какую скорость считать используемой на интерфейсе.)

Другие сравнения протокола IGP

Протоколы IGP можно сравнить и иначе. Тем не менее некоторые темы требуют более глубокого знания определенных протоколов маршрутизации и других рассматриваемых в этой книге средств. Данный раздел знакомит еще с несколькими пунктами сравнения, а детали мы рассмотрим позже.

В первую очередь, протоколы маршрутизации различаются на основании того, являются ли они бесклассовыми протоколами маршрутизации (а следовательно, поддерживающими маски VLSM) или классовыми (не поддерживающими маски VLSM). Маски VLSM обсуждаются подробно в главе 20. *Бесклассовые протоколы маршрутизации* (classless routing protocol) поддерживают маски VLSM, посылая сообщения об обновлении с масками подсети, в то время как прежние *классовые протоколы маршрутизации* (classful routing protocol) не посылают маски в сообщениях анонса маршрутизации.

Протоколы маршрутизации отличаются также способностью поддерживать суммирование маршрутов. Суммирование маршрутов обсуждается в главе 21, а пока вам достаточно знать, что современные сети способны использовать суммирование маршрутов вручную, а все упомянутые в этой книге протоколы маршрутизации IGP, кроме действительно устаревшей версии 1 протокола RIP (RIP 1), поддерживают суммирование маршрутов вручную.

Краткое сравнение протоколов IGP приведено в табл. 17.2.

Ключевая тема **Таблица 17.2. Сравнение внутренних протоколов маршрутизации IP**

Средство	RIP-1	RIP-2	EIGRP	OSPF	IS-IS
Бесклассовый, пересылающий маски в обновлениях, поддерживающих VLSM	Нет	Да	Да	Да	Да
Алгоритм (DV, расширенный DV, LS)	DV	DV	Расширенный DV	LS	LS
Поддержка суммирования вручную	Нет	Да	Да	Да	Да
Собственность Cisco	Нет	Нет	Да *	Нет	Нет
Анонсы маршрутизации поступают на многоадресатный IP-адрес	Нет	Да	Да	Да	—
Конвергенция	Медленная	Медленная	Быстрая	Быстрая	Быстрая

* Хотя компания Cisco создала протокол EIGRP и поддерживала его как собственный на протяжении многих лет, на момент публикации этой книги предполагалось оформить протокол EIGRP документом RFC. Это позволит и другим производителям реализовать протокол EIGRP, в то время как компания Cisco сохранит права на него.

Административное расстояние

Большинство компаний и организаций используют единый протокол маршрутизации. Однако в некоторых случаях компания должна использовать несколько протоколов маршрутизации. Например, если две компании соединяют свои сети, чтобы обмениваться информацией, они должны обмениваться некой информацией о мар-

шрутизации. Если одна компания использует протокол OSPF, а другая протокол EIGRP по крайней мере на одном маршрутизаторе, то поддерживать придется оба протокола — OSPF и EIGRP. Впоследствии этот маршрутизатор может получать маршруты по протоколу OSPF, а анонсировать их по протоколу EIGRP, и наоборот, осуществляя *перераспределение маршрута* (route redistribution).

В зависимости от топологии сети эти два протокола маршрутизации могли бы изучить маршруты к тем же подсетям. Когда один протокол маршрутизации изучает несколько маршрутов к той же подсети, лучший из них определяет метрика. Но когда два разных протокола маршрутизации изучают маршруты к той же подсети, операционная система IOS не может сравнить метрики, поскольку у каждого протокола маршрутизации они основаны на разной информации. Например, протокол OSPF мог бы изучить маршрут к подсети 10.1.1.0 с метрикой 101, а протокол EIGRP мог бы изучить маршрут к той же подсети 10.1.1.0 с метрикой 2 195 416, и этот маршрут может быть лучшим маршрутом или не быть. Для сравнения этих двух метрик просто нет никакого основания.

Когда операционная система IOS должна выбирать между маршрутами, изученными разными протоколами маршрутизации, она использует концепцию *административного расстояния* (administrative distance). Административное расстояние — это числовое представление достоверности протокола маршрутизации на одном маршрутизаторе. Чем ниже это значение, тем лучше, тем достоверней протокол маршрутизации. Например, стандартное административное расстояние для протокола RIP составляет 120, для протокола OSPF — 110, а для протокола EIGRP — 90. При использовании протоколов OSPF и EIGRP маршрутизатор будет верить маршруту по протоколу EIGRP, а не OSPF (по крайней мере, изначально). Значения административного расстояния настраиваются на маршрутизаторе индивидуально, и они не обмениваются ими друг с другом. В табл. 17.3 приведены различные источники информации о маршрутизации, а также стандартные административные расстояния.

Таблица 17.3. Стандартные административные расстояния

Ключевая тема

Тип маршрута	Административное расстояние
Подключенный (connected)	0
Статический (static)	1
BGP (внешние маршруты)	20
EIGRP (внутренние маршруты)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EIGRP (внешние маршруты)	170
BGP (внутренние маршруты)	200
Не используется	255

ВНИМАНИЕ!

Команда `show ip route` выводит административное расстояние каждого маршрута как первое из двух чисел в скобках. Второе число в скобках — это метрика.

В таблице представлены стандартные значения административного расстояния, но они могут быть перенастроены как для специфического протокола маршрутизации, так и для специфического и даже статического маршрута. Например, команда `ip route 10.1.3.0 255.255.255.0 10.1.130.253` определяет статический маршрут со стандартным административным расстоянием 1, а команда `ip route 10.1.3.0 255.255.255.0 10.1.130.253 210` определяет тот же статический маршрут с административным расстоянием 210. Так можно создать статический маршрут, используемый только тогда, когда протокол маршрутизации не находит другой маршрут, — достаточно предоставить статический маршрут с более высоким административным расстоянием.

Понятие протокола маршрутизации по состоянию канала OSPF

Протоколы маршрутизации обеспечивают в основном обмен информацией, позволяющий маршрутизаторам изучать маршруты. Маршрутизаторы изучают информацию о подсетях, маршрутах к этим подсетям и о метриках, свидетельствующих о том, насколько хорош каждый маршрут по сравнению с другими. Протокол маршрутизации позволяет также выбрать оптимальный в настоящее время маршрут к каждой подсети и построить таблицу маршрутизации IP.

В данном разделе будет продолжено рассмотрение концепций протоколов маршрутизации, при этом основное внимание уделяется протоколам состояния канала, а именно протоколу OSPF. Раздел начинается с обсуждения того, как маршрутизаторы изучают информацию по протоколу OSPF и выбирают маршруты, добавляемые в таблицу маршрутизации. Далее обсуждение возвращается к фундаментальной части процесса: тому, как маршрутизаторы используют и устанавливают соседские отношения по протоколу OSPF, прежде чем приступить к обмену маршрутами и информацией о маршрутизации. Завершается раздел описанием масштабирования проекта корпоративной сети и его влияния на увеличение размеров работы такого протокола состояния канала как OSPF.

Создание баз LSDB и маршрутов IP

Протоколы состояния канала создают маршруты IP в ходе нескольких этапов. Сначала маршрутизаторы совместно выясняют достаточно много информации о сети: ее маршрутизаторы, каналы связи, IP-адреса, информацию о состоянии и т.д. Затем они рассылают эту информацию, чтобы все маршрутизаторы знали ее. На данный момент любой маршрутизатор может вычислять маршруты ко всем подсетям со своей точки зрения.

Информация о топологии и анонсы LSA

Используя протоколы маршрутизации по состоянию канала, маршрутизаторы должны анонсировать практически все подробности об объединенной сети для всех остальных маршрутизаторов. Наконец, процесс *лавинной рассылки* (flooding) доставляет информацию всем маршрутизаторам в объединенной сети, чтобы у каждого из них была та же информация об объединенной сети.

Открытый протокол поиска первого кратчайшего маршрута (OSPF) — наиболее популярный протокол маршрутизации IP по состоянию канала — организует информацию о топологии, используя *анонсы состояния канала* (Link-State Advertisement —

LSA) и *базу данных состояний каналов* (Link-State Database — LSDB). Концепция представлена на рис. 17.4. Каждый анонс LSA — это структура данных, содержащая немного специфической информации о топологии сети; база LSDB — это просто коллекция всех анонсов LSA, известных маршрутизатору. Команда `show ip ospf database`, отданная в интерфейсе командной строки маршрутизатора, использующего протокол OSPF, отобразит базу LSDB на данном маршрутизаторе, а также часть информации в каждом анонсе LSA в базе LSDB.

Рис. 17.4. Взаимоотношение анонсов LSA и базы LSDB

Рис. 17.5 дает общее представление о процессе лавинной рассылки, осуществляемой маршрутизатором R8 для передачи анонса LSA. Анонс LSA маршрутизатора R8 описывает сам маршрутизатор и существующую подсеть 172.16.3.0/24 (справа на рисунке). (Обратите внимание, что на рис. 17.5 показана только часть информации о маршрутизаторе R8 в анонсе LSA.)

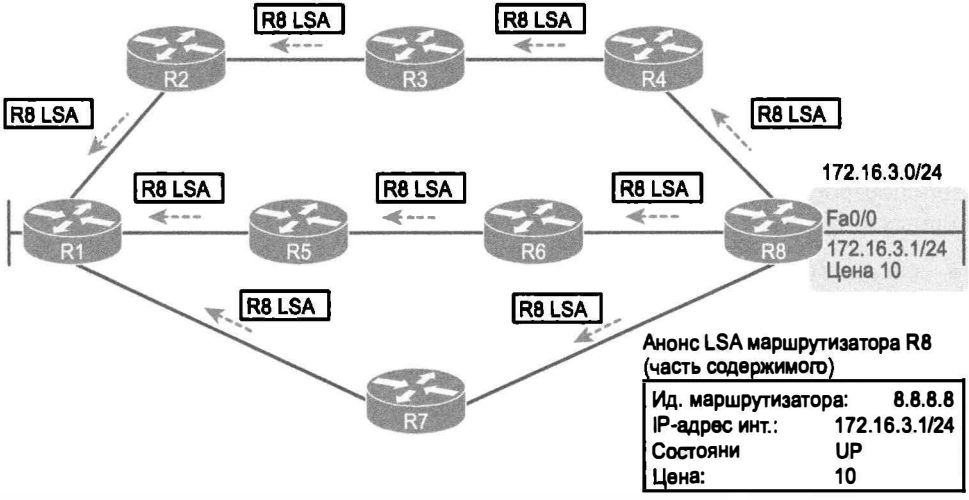


Рис. 17.5. Лавинная рассылка анонса LSA с использованием протокола маршрутизации по состоянию канала

На рис. 17.5 приведен довольно простой пример процесса лавинной рассылки, когда маршрутизатор R8 публикует анонс LSA о себе, а другие маршрутизаторы перенаправляют его до тех пор, пока у каждого маршрутизатора не будет по экземпляру. Процесс лавинной рассылки предотвращает циклическую передачу, чтобы анонсы

LSA не заняли весь трафик. Обычно перед передачей анонса LSA маршрутизатор спрашивает соседа: "У вас уже есть этот анонс LSA?", и если он есть, то передача отменяется.

Иногда маршрутизаторы повторяют лавинную рассылку отдельных анонсов LSA. Это происходит при изменении маршрутной информации, например, при отказе или восстановлении канала связи. Анонсы LSA повторно рассылаются также по истечении таймера устаревания (стандартно 30 минут).

Поиск наилучших маршрутов с помощью алгоритма Дейкстры

В результате лавинной рассылки анонса состояния канала на каждом маршрутизаторе будет одинаковый экземпляр базы LSDB, но в таблицу маршрутизации IP этот процесс наилучшие маршруты не добавит. Хотя информация в базе LSDB невероятно подробна и полезна, в ней не значится оптимальный маршрут от каждого маршрутизатора к каждому получателю.

Для получения маршрутов маршрутизаторы осуществляют математические вычисления. К счастью, ни мне ни вам не нужно знать математику столь глубоко! Тем не менее для обработки базы LSDB все протоколы состояния канала используют математический *алгоритм поиска первого кратчайшего пути Дейкстры* (Dijkstra Shortest Path First — SPF). Этот алгоритм анализирует (математически) базу LSDB и создает маршруты, добавляемые локальными маршрутизаторами в таблицы маршрутизации IP. Маршруты включают номер подсети и маску, исходящий интерфейс и IP-адрес следующего транзитного маршрутизатора.

Более подробная информация о процессе SPF приведена во втором томе книги. По крайней мере, ее достаточно для планирования конфигурации OSPF. Там, в частности, рассматривается выбор метрик OSPF, влияющий на принимаемые алгоритмом SPF решения и позволяющий сетевому инженеру воздействовать на выбор маршрутизатором наилучшего маршрута.

Использование соседских отношений

Для построения маршрутов протокол OSPF использует внутренне операции трех основных категорий.

- *Соседи* (neighbors). Отношения между двумя маршрутизаторами, соединенными тем же каналом связи. Соседние маршрутизаторы способны обмениваться базами LSDB.
- *Обмен базами данных* (database exchange). Процесс пересылки анонсов LSA соседям, чтобы все маршрутизаторы изучили этот анонс.
- *Добавление наилучших маршрутов* (adding the best route). Процесс вычисления по локальной копии базы LSDB наилучших маршрутов и добавления их в таблицу маршрутизации IPv4 на каждом поддерживающем алгоритм SPF маршрутизаторе.

Последние два элемента списка уже обсуждались выше, но не соседские отношения с другими маршрутизаторами. Фактически большая часть процесса проверки, поиска и устранения неисправностей вращается вокруг соседских отношений OSPF. Их основные принципы обсуждаются в данном разделе.

Основы соседских отношений OSPF

Соседи OSPF — это маршрутизаторы, находящиеся на том же канале связи и использующие протокол OSPF. В соответствии с обсуждаемой в этой книге технологией канала связи два маршрутизатора, соединенных тем же каналом VLAN, являются соседями OSPF; соседями являются также два маршрутизатора на концах последовательного канала связи.

Чтобы стать соседями OSPF, двум маршрутизаторам недостаточно находиться на том же канале связи, они должны обменяться сообщениями OSPF и согласиться стать соседями. Для этого маршрутизаторы посылают сообщение Hello OSPF, представляясь соседу. С учетом совместимости параметров OSPF у двух соседей их отношения будут отображены в выводе команды `show ip ospf neighbors`.

Соседские отношения OSPF позволяют также установить, что сосед прямо сейчас может не быть наилучшим выбором для перенаправления пакета. Предположим, что маршрутизаторы R1 и R2 имеют соседские отношения, изучают анонсы LSA и вычисляют маршруты передачи пакетов через другой маршрутизатор. Несколько месяцев спустя маршрутизатор R1 обращает внимание на отсутствие соседских отношений с маршрутизатором R2, и это заставляет его реагировать: он повторно рассылает анонсы LSA, полагаясь на наличие канала связи между маршрутизаторами R1 и R2. Затем маршрутизатор R1 запускает алгоритм SPF, чтобы повторно вычислить собственные маршруты.

И наконец, модель соседей OSPF обеспечивает новым маршрутизаторам динамическое обнаружение. Это означает, что новые маршрутизаторы могут быть добавлены в сеть без обязательной перенастройки каждого маршрутизатора. Вместо этого конфигурация включает протокол OSPF на интерфейсах маршрутизатора, а затем он начинает реагировать на все сообщения Hello от новых соседей, когда они появляются.

Общение соседей и изучение их идентификаторов маршрутизатора

Процесс обработки сообщений Hello OSPF, в ходе которого формируются новые соседские отношения, немного похож на то, как люди знакомятся с соседями при переезде в новый дом. Встретив нового соседа, люди обычно здороваются и представляются (говорят hello и называют свои имена). После короткого разговора создается первое впечатление о соседе, будет ли с ним интересно побеседовать иногда или только здороваться при встрече.

Точно так же процесс OSPF начинается с сообщений Hello OSPF. Эти сообщения содержат *идентификаторы маршрутизатора* (Router ID — RID) — уникальные имена или идентификаторы каждого маршрутизатора для протокола OSPF. И наконец, протокол OSPF проверяет информацию в сообщениях Hello и удостоверяется, что эти два маршрутизатора могут стать соседями.

Идентификатор RID OSPF — это 32-разрядное число. В результате большинство команд выводит их в десятичном представлении с разделительными точками. Кроме того, стандартно операционная система IOS назначает идентификаторы RID OSPF на основании IPv4-адреса интерфейса, поскольку это тоже вполне подходящее 32-разрядное число. Однако идентификатор RID OSPF может быть задан непосредственно, как описано далее.

Как только маршрутизатор выбрал идентификаторы RID OSPF и включил интерфейсы, он готов знакомиться с соседями. Маршрутизаторы OSPF могут быть соседями, если они подключены к той же подсети (хотя есть и другие частные случаи, не рассматриваемые на экзаменах CCENT и CCNA). Для обнаружения других поддерживающих протокол OSPF маршрутизаторов со всех интерфейсов рассылаются многоадресатные пакеты Hello OSPF и ожидаются ответы на них от других маршрутизаторов, подключенных к этим интерфейсам. Концептуально это представлено на рис. 17.6.

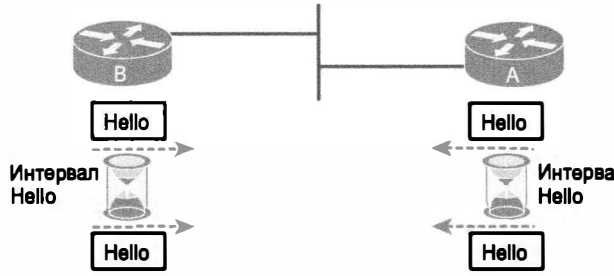


Рис. 17.6. Пакеты Hello OSPF

Маршрутизаторы A и B посылают сообщения Hello в сеть LAN. Они продолжают регулярно посылать сообщения Hello с интервалом, заданным параметром Hello Timer. Само сообщение Hello имеет следующие свойства.

- Сообщение Hello предваряется заголовком пакета IP с типом протокола 89.
- Пакеты Hello посылают на многоадресатный IP-адрес 224.0.0.5, предназначенный для всех маршрутизаторов, поддерживающих протокол OSPF.
- Маршрутизаторы OSPF получают пакеты Hello, посылаемые на многоадресатный IP-адрес 224.0.0.5, и узнают из них о новых соседях.

Сообщения Hello содержат множество параметров OSPF. Они позволяют каждому маршрутизатору узнать много подробностей о потенциальном соседе, включая информацию о том, будут ли эти два маршрутизатора соседями. Например, два маршрутизатора OSPF не будут соседями, если IPv4-адреса их интерфейсов будут находиться в разных подсетях. Только то, что два маршрутизатора получают сообщения Hello друг от друга, еще не означает, что эти два маршрутизатора станут соседями. Но если маршрутизаторы становятся соседями, то они начинают обмениваться своими базами LSDB, а затем вычисляют новые маршруты IP.

ВНИМАНИЕ!

Поиск и устранение неисправностей протокола маршрутизации с изрядным количеством подробностей о причинах того, почему маршрутизаторы OSPF и EIGRP не смогут стать соседями, описаны во втором томе книги.

Масштабирование OSPF за счет иерархического проекта

Протокол OSPF может быть применен в сетях, проблемы проекта которых продуманы не очень глубоко. Достаточно только включить протокол OSPF на всех маршрутизаторах, и он заработает! Однако в больших сетях инженер должен обду-

мать и спланировать использование средств протокола OSPF, чтобы обеспечить хорошее масштабирование. Например, проект OSPF на рис. 17.7 использует единую область OSPF, поскольку эта небольшая объединенная сеть не нуждается в преимуществах масштабирования областей OSPF.

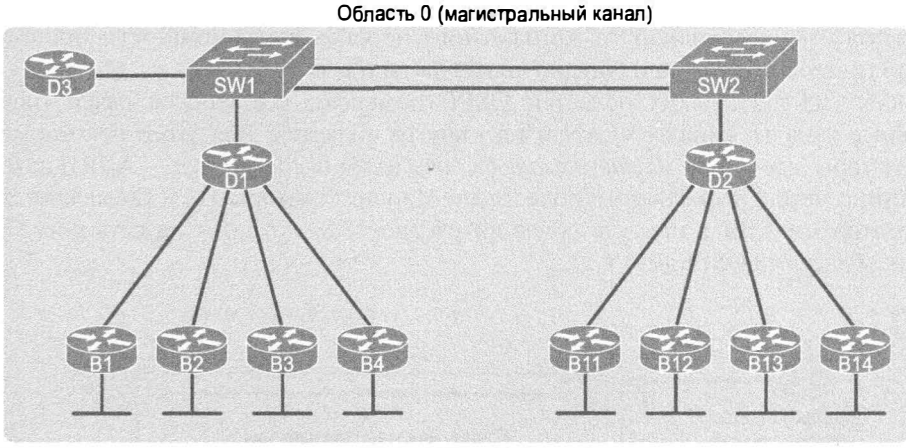


Рис. 17.7. Единая область OSPF

Единая область OSPF хорошо подходит для небольших объединенных сетей, как показано на рис. 17.7. Конфигурация проста, а некоторые из скрытых деталей работы протокола OSPF остаются простыми. Фактически в небольшой объединенной сети можно просто включить протокол OSPF на всех интерфейсах в той же области и игнорировать концепцию области OSPF.

Теперь вместо сети с одиннадцатью маршрутизаторами вообразите сеть с 900 маршрутизаторами и несколькими тысячами подсетей. При таком размере сети работа сложного алгоритма SPF способна существенно замедлить конвергенцию, постольку каждый маршрутизатор потратит некоторое время на выполнение всех математических действий. Кроме того, маршрутизаторам могло бы не хватить памяти. В итоге могут возникнуть следующие проблемы.

- Большая топологическая база данных занимает больше памяти на каждом маршрутизаторе.
- Обработка большей топологической базы данных алгоритмом SPF естественно требует большей мощности, причем в экспоненциальной зависимости от размера базы.
- Изменение состояния одного интерфейса (включен или выключен) вынуждает каждый маршрутизатор снова запускать алгоритм SPF!

Используя области, протокол OSPF позволяет разделить большую и сложную задачу выполнения алгоритма SPF на большой базе LSDB. Инженер помещает одну часть каналов в одну область, а другие в другую область, в третью и т.д. В результате протокол OSPF создает меньшие базы LSDB областей, а не одну огромную базу LSDB для всех каналов связи и маршрутизаторов в объединенной сети. При меньших топологических базах данных маршрутизаторы используют меньше памяти и быстрее осуществляют ее обработку алгоритмом SPF.

Несмотря на то что в этом контексте и нет точного определения "больших" сетей, если в сети больше нескольких дюжин маршрутизаторов, то использование нескольких областей будет скорее преимуществом, чем недостатком. (В некоторых документах как максимум для одной области упоминается 50 маршрутизаторов.) Но обратите внимание, что эти количества маршрутизаторов очень общие. Они в значительной степени зависят от конкретного проекта сети, мощности процессоров маршрутизаторов, объема их оперативной памяти и т.д.

Проект из нескольких областей OSPF помещает все каналы связи (последовательные каналы, каналы VLAN и т.д.) внутрь областей. Для этого некоторые маршрутизаторы (*граничные маршрутизаторы зоны* (Area Border Router — ABR)) находятся на границе между несколькими областями. Маршрутизаторы D1 и D2 являются маршрутизаторами ABR в проекте областей на рис. 17.8, похожем на сеть рис. 17.7, но с тремя областями OSPF (0, 1 и 2).

Ключевая тема

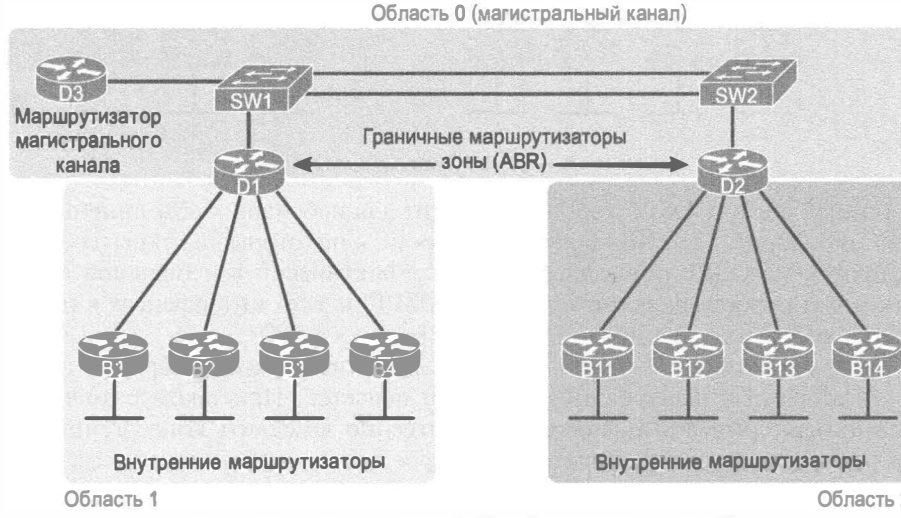


Рис. 17.8. Терминология множества областей OSPF

Хотя на рис. 17.8 представлен типовой проект областей и продемонстрирована связанная с ним терминология, ни польза, ни преимущества областей здесь не очевидны. При использовании областей OSPF алгоритм SPF игнорирует подробности топологии в других областях. Например, выполняя сложные математические вычисления по алгоритму SPF на маршрутизаторе B1 (область 1), протокол OSPF игнорирует информацию о топологии в областях 0 и 2. Таким образом, у каждого маршрутизатора будет гораздо меньше работы, алгоритм SPF значительно быстрее закончит свою работу по поиску наилучших в настоящее время маршрутов OSPF.

Работа алгоритма SPF и типы анонсов LSA подробней рассматриваются во втором томе книги. А пока вам достаточно знать, что одиночные области лучше подходят для меньших сетей, а проекты с несколькими областями решают проблемы масштабирования, возникающие при разрастании сети.

Конфигурация OSPF

Для настройки OSPF необходимо лишь несколько этапов, но необязательных этапов может быть множество. После выбора проекта OSPF (в больших объединенных сетях IP эта задача может быть сложной) настройка может быть очень простой: включение протокола OSPF на каждом интерфейсе маршрутизатора и помещение этого интерфейса в соответствующую область OSPF.

В этом разделе приведено несколько примеров конфигурации объединенной сети OSPF с одной областью. После этих примеров рассматривается несколько дополнительных, необязательных параметров конфигурации. Ниже описаны рассматриваемые в данной главе этапы настройки OSPF, а также краткий перечень необходимых команд.

Этапы настройки протокола OSPF

Ключевая тема

Этап 1 Перейдите в режим конфигурации определенного процесса OSPF, используя глобальную команду `router ospf` *`идентификатор_процесса`*

Этап 2 (Необязательно.) Настройка идентификатора маршрутизатора OSPF:

A. Настройка подкоманды маршрутизатора `router-id` *`значение_идентификатора`*.

B. Настройка IP-адреса на *петлевом интерфейсе* (loopback interface)

Этап 3 Настройка одной или нескольких подкоманд маршрутизатора `network` *`ip-адрес шаблон_маски`* `area` *`идентификатор_области`* со всеми соответствующими интерфейсами, добавляемыми к указанным областям

Для большей наглядности на рис. 17.9 представлены процесс настройки протокола OSPFv2, а также отношения между командами конфигурации OSPF. Обратите внимание, что настройка создает процесс маршрутизации в одной части конфигурации, а затем косвенно включает протокол OSPF на каждом интерфейсе. Конфигурация не перечисляет интерфейсы, на которых включен протокол OSPF, вместо этого операционная система IOS применяет логический процесс сравнения команды OSPF `network` с командой интерфейса `ip address`. Более подробная информация об этой логике приведена в следующем примере.

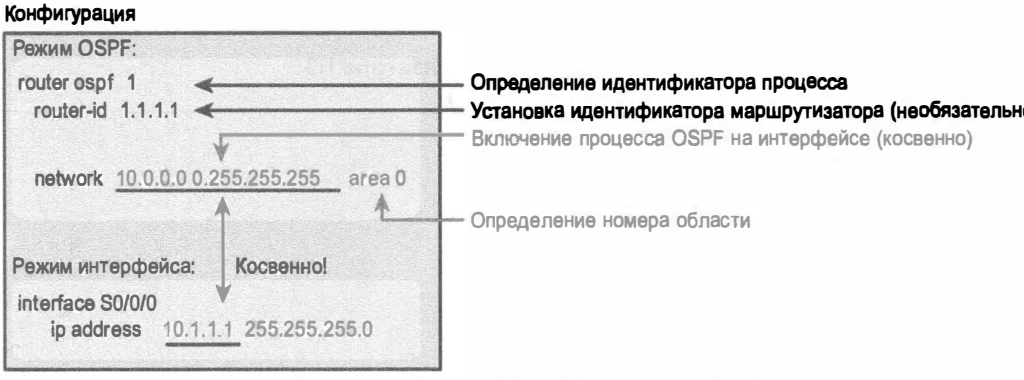


Рис. 17.9. Организация настройки OSPFv2

Настройка одиночной области OSPF

На рис. 17.10 приведен пример сети, используемой для конфигурации OSPF. Все каналы связи находятся в области 0. Здесь есть четыре маршрутизатора, каждый из которых подключен к одной или двум локальным сетям. Но обратите внимание, что маршрутизаторы R3 и R4 (см. рис. 17.10, *сверху*) подключены к тем же двум VLAN (подсетям), а следовательно, имеют соседские отношения друг с другом по каждой из этих VLAN.

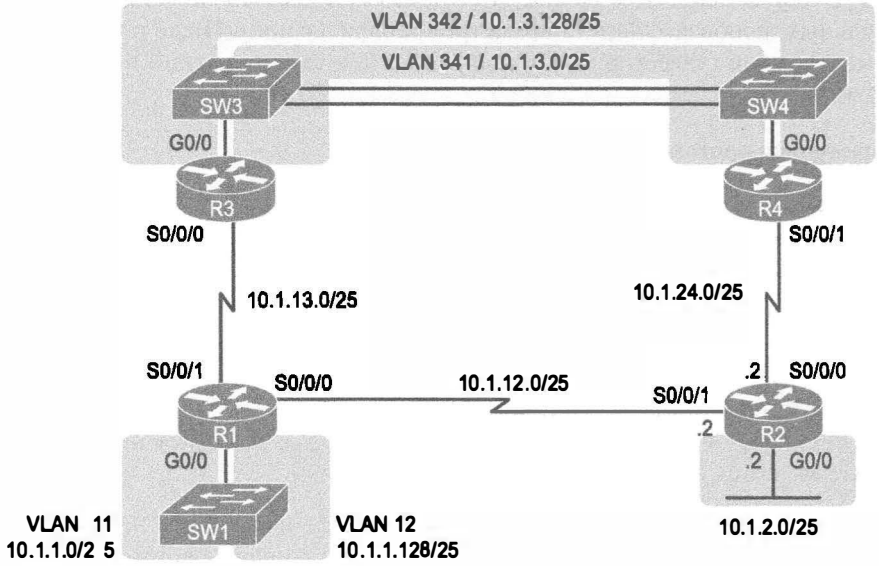


Рис. 17.10. Пример сети для конфигурации OSPF с одной областью

Прежде чем переходить к детальному рассмотрению OSPF, обратимся к примеру 17.1 конфигурации IPv4-адресации на маршрутизаторе R3. В конфигурации маршрутизатора R3 каждому интерфейсу присвоен IP-адрес, а на интерфейсе G0/0 разрешено магистральное соединение 802.1Q. (Хотя здесь и не показано, на коммутаторе SW3 настроено магистральное соединение с другой стороной этого канала связи Ethernet.)

Пример 17.1. Настройка IPv4-адресов на маршрутизаторе R3 (включающем магистральное соединение VLAN)

```
interface gigabitethernet 0/0.341
  encapsulation dot1q 341
  ip address 10.1.3.1 255.255.255.128
!
interface gigabitethernet 0/0.342
  encapsulation dot1q 342
  ip address 10.1.3.129 255.255.255.128
!
interface serial 0/0/0
  ip address 10.1.13.3 255.255.255.128
```

Изначальная конфигурация одиночной области на маршрутизаторе R3, как показано в примере 17.2, включает протокол OSPF на всех интерфейсах, представленных на рис. 17.9. Сначала глобальная команда `router ospf 1` переводит пользователя в режим конфигурации OSPF и устанавливает *идентификатор процесса* (process ID) OSPF. Это число должно быть уникально только на локальном маршрутизаторе, применение различных идентификаторов процесса позволяет маршрутизатору поддерживать несколько процессов OSPF в одном маршрутизаторе. (Команда `router` использует идентификатор процесса для различения процессов.) Идентификатор процесса не должен соответствовать каждому маршрутизатору и может быть любым целым числом от 1 до 65 535.

Пример 17.2. Настройка одиночной области OSPF на маршрутизаторе R3 с использованием одной команды `network`

```
router ospf 1
  network 10.0.0.0 0.255.255.255 area 0
```

В общем случае, а не как в данном конкретном примере, команда `network` указывает маршрутизатору найти свои локальные интерфейсы, соответствующие первым двум параметрам команды `network`. Затем для каждого соответствующего интерфейса маршрутизатор включает протокол OSPF на этих интерфейсах, обнаруживает соседей, устанавливает соседские отношения и присваивает интерфейс области, указанной командой `network`.

Конкретная команда в примере 17.2 назначает все соответствующие интерфейсы области 0. Однако первые два параметра, *ip-адрес* и *шаблон_маски*, имеющие значения 10.0.0.0 и 0.255.255.255, требуют некоторых пояснений. В данном случае команде соответствуют все три интерфейса, представленные для маршрутизатора R3 (в следующем разделе объясняется, почему).

Соответствие команде OSPF `network`

Команда OSPF `network` сравнивает свой первый параметр с каждым IP-адресом интерфейса на локальном маршрутизаторе, пытаясь найти соответствие. Но вместо сравнения всех номеров в команде `network` со всеми IPv4-адресами интерфейсов маршрутизатор может сравнивать лишь подмножество октетов на основании шаблона маски следующим образом.

Шаблон маски OSPF и его значение

`Wildcard 0.0.0.0`. Сравнивает все 4 октета (т.е. числа должны совпадать точно).

`Wildcard 0.0.0.255`. Сравнивает только первые 3 октета. Последний октет при сравнении игнорируется.

`Wildcard 0.0.255.255`. Сравнивает только первые 2 октета. Последние 2 октета при сравнении игнорируются.

`Wildcard 0.255.255.255`. Сравнивает только первый октет. Последние 3 октета при сравнении игнорируются.

`Wildcard 255.255.255.255`. Не сравнивает ничего. Этот шаблон маски означает соответствие команде `network` любых адресов.

Значение 0 в октете шаблона маски указывает операционной системе IOS сравнивать этот октет на соответствие, а значение 255 — требует игнорировать этот октет при сравнении чисел.

Благодаря шаблону маски команда `network` обладает высокой гибкостью. Например, на маршрутизаторе R3 может быть применено множество команд `network`, одни из которых соответствуют всем интерфейсам, а другие — лишь подмножеству интерфейсов. В табл. 17.4 приведено несколько примеров с примечаниями.

Таблица 17.4. Пример команды `network` на маршрутизаторе R3 с ожидаемыми результатами

Команда	Логика команды	Соответствующие интерфейсы
`network 10.1.0.0 0.0.255.255`	Соответствуют IP-адреса интерфейсов, начинающиеся на 10.1	G0/0.341 G0/0.342 S0/0/0
`network 10.0.0.0 0.255.255.255`	Соответствуют IP-адреса интерфейсов, начинающиеся на 10	G0/0.341 G0/0.342 S0/0/0
`network 0.0.0.0 255.255.255.255`	Соответствуют все IP-адреса интерфейсов	G0/0.341 G0/0.342 S0/0/0
`network 10.1.13.0 0.0.255.255`	Соответствуют IP-адреса интерфейсов, начинающиеся на 10.1.13	S0/0/0
`network 10.1.3.1 0.0.0.0`	Соответствует только один IP-адрес: 10.1.3.1	G0/0.341

Шаблон маски определяет для локального маршрутизатора правила поиска соответствующих интерфейсов. Пример 17.2 демонстрирует использование команды `network 10.0.0.0 0.255.255.255 area 0` на маршрутизаторе R3. Маршрутизаторы R1 и R2 в той же объединенной сети использовали бы конфигурацию, представленную в примере 17.3, с двумя другими шаблонами маски. На обоих маршрутизаторах протокол OSPF включается на всех интерфейсах, представленных на рис. 17.10.

Пример 17.3. Конфигурация OSPF на маршрутизаторах R1 и R2

```
! Далее конфигурация R1 - одна команда network включает протокол OSPF
! на всех трех интерфейсах
router ospf 1
  network 10.1.0.0 0.0.255.255 area 0

! Далее конфигурация R2 - по одной команде network на каждый интерфейс
router ospf 1
  network 10.1.12.2 0.0.0.0 area 0
  network 10.1.24.2 0.0.0.0 area 0
  network 10.1.2.2 0.0.0.0 area 0
```

И наконец, вполне возможны и другие значения шаблонов маски, что позволяет осуществлять сравнение конкретных битов 32-разрядных чисел. Более подробная информация о шаблонах и других возможностях масок приведена в главе 22.

ВНИМАНИЕ!

Для первого параметра (адреса) команда `network` использует другое соглашение: если октет будет проигнорирован в связи со значением 255 октета шаблона маски, то октетом параметра адреса будет 0. Однако операционная система IOS вполне примет команду `network`, нарушающую это правило, но затем изменит этот октет адреса на 0, прежде чем поместить его в файл текущей конфигурации. Например, операционная система IOS изменит введенную команду `network 1.2.3.4 0.0.255.255` на `network 1.2.0.0 0.0.255.255`.

Проверка OSPF

Как уже упоминалось, маршрутизаторы OSPF используют процесс из трех этапов. Сначала они устанавливают соседские отношения, затем создают и рассылают анонсы LSA, чтобы у каждого маршрутизатора в той же области была одинаковая копия базы LSDB. И наконец, каждый маршрутизатор независимо вычисляет собственные маршруты IP и добавляет их в таблицу маршрутизации.

Команды `show ip ospf neighbor`, `show ip ospf database` и `show ip route` отображают информацию о каждом из трех этапов соответственно. Для проверки состояния OSPF их можно использовать в той же последовательности. Либо можно просто просмотреть таблицу маршрутизации IP и, если маршруты выглядят правильно, протокол OSPF, вероятно, работает правильно.

Сначала рассмотрим список соседей, известных маршрутизатору R3. У маршрутизатора R3 должны быть соседские отношения с маршрутизатором R1 по последовательному каналу связи. У него также есть соседские отношения с маршрутизатором R4 по двум разным каналам VLAN, к которыми подключены оба маршрутизатора. Все три соседа представлены в примере 17.4.

Пример 17.4. Соседи OSPF на маршрутизаторе R3 согласно рис. 17.9

```
R3# show ip ospf neighbor

Neighbor ID  Pri  State     Dead Time  Address     Interface
1.1.1.1        0  FULL/ -   00:00:33   10.1.13.1   Serial0/0/0
10.1.24.4      1  FULL/DR   00:00:35   10.1.3.130 GigabitEthernet0/0.342
10.1.24.4      1  FULL/DR   00:00:36   10.1.3.4   GigabitEthernet0/0.341
```

Вывод отражает несколько важных фактов. Заголовки вывода удобней просматривать слева направо.

`Interface` (интерфейс). Интерфейс локального маршрутизатора, подключенный к соседу. Например, первый сосед в списке доступен через интерфейс S0/0/0 маршрутизатора R3.

`Address` (адрес). IP-адрес соседа на данном канале связи. Для первого соседа, маршрутизатора R1, это IP-адрес 10.1.13.1.

`State` (состояние). Хотя в этой главе обсуждалось много возможных состояний, в данном случае `FULL` означает правильное и полностью рабочее состояние.

`Neighbor ID` (идентификатор соседа). Идентификатор соседнего маршрутизатора.

В примере 17.5 приведено содержимое базы LSDB на маршрутизаторе R3. Интересен тот факт, что при правильной работе протокола OSPF в объединенной сети

содержимое базы LSDB у всех маршрутизаторов в одной области будет одинаковым. Поэтому команда `show ip ospf database` в примере 17.5 должна отобразить точно такую же информацию, независимо от любого из этих четырех маршрутизаторов.

Пример 17.5. База данных OSPF на маршрутизаторе R3 согласно рис. 17.10

```
R3# show ip ospf database

       OSPF Router with ID (10.1.13.3) (Process ID 1)

                Router Link States (Area 0)

Link ID    ADV Router  Age  Seq#       Checksum  Link count
1.1.1.1    1.1.1.1     498  0x80000006 0x002294  6
2.2.2.2    2.2.2.2     497  0x80000004 0x00E8C6  5
10.1.13.3  10.1.13.3   450  0x80000003 0x001043  4
10.1.24.4  10.1.24.4   451  0x80000003 0x009D7E  4

                Net Link States (Area 0)

Link ID    ADV Router  Age  Seq#       Checksum
10.1.3.4   10.1.24.4   451  0x80000001 0x0045F8
10.1.3.130 10.1.24.4   451  0x80000001 0x00546B
```

Не обращайте пока внимания на конкретные данные в выводе этой команды. Однако отметьте, что база LSDB должна насчитывать по одному разделу "`Router Link State`" для каждого из этих четырех маршрутизаторов, как выделено в примере.

Далее, в примере 17.6, приведена таблица IPv4 маршрутизации маршрутизатора R3, выведенная командой `show ip route`. Здесь перечислены как маршруты OSPF, так и подключенные маршруты. Если вернуться к рис. 17.10, то можно найти подсети, которые не подключены локально к маршрутизатору R3. Маршруты к ним можно найти в выводе примера 17.5.

Пример 17.6. Маршруты IPv4, добавленные протоколом OSPF на маршрутизатор R3, согласно рис. 17.10

```
R3# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
! Предыдущие строки опущены для краткости
   10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
O   10.1.1.0/25 [110/65] via 10.1.13.1, 00:13:28, Serial0/0/0
O   10.1.1.128/25 [110/65] via 10.1.13.1, 00:13:28, Serial0/0/0
O   10.1.2.0/25 [110/66] via 10.1.3.130, 00:12:41, GigabitEthernet0/0.342
                [110/66] via 10.1.3.4, 00:12:41, GigabitEthernet0/0.341
C   10.1.3.0/25 is directly connected, GigabitEthernet0/0.341
L   10.1.3.1/32 is directly connected, GigabitEthernet0/0.341
C   10.1.3.128/25 is directly connected, GigabitEthernet0/0.342
L   10.1.3.129/32 is directly connected, GigabitEthernet0/0.342
O   10.1.12.0/25 [110/128] via 10.1.13.1, 00:13:28, Serial0/0/0
C   10.1.13.0/25 is directly connected, Serial0/0/0
L   10.1.13.3/32 is directly connected, Serial0/0/0
O   10.1.24.0/25
          [110/65] via 10.1.3.130, 00:12:41, GigabitEthernet0/0.342
          [110/65] via 10.1.3.4, 00:12:41, GigabitEthernet0/0.341
```

Сначала обратите внимание на общие идеи, подтвержденные данным выводом. Код "O" слева означает маршрут, изученный по протоколу OSPF. В выводе перечислено пять таких маршрутов IP. На рисунке приведено пять подсетей, не подключенных к маршрутизатору R3. Быстрый взгляд на маршруты OSPF, по сравнению с маршрутами, отличными от подключенных, позволяет удостовериться, что протокол OSPF изучил все маршруты.

Затем обратите внимание на первый маршрут (к подсети 10.1.1.0/25). В нем перечислены идентификатор подсети и маска, идентифицирующие подсеть. Он выводит также два числа в скобках. Первое, 110, — это административное расстояние маршрута. Все маршруты OSPF в этом примере используют стандартное значение 110. Второе число, 65, — это метрика OSPF для данного маршрута.

Кроме того, для быстрой проверки работы любого протокола маршрутизации весьма популярна также команда `show ip protocols`. Она выводит группы сообщений для каждого выполняющегося на маршрутизаторе протокола маршрутизации. Пример 17.7 демонстрирует это для маршрутизатора R3.

Пример 17.7. Команда `show ip protocols` на маршрутизаторе R3

```
R3# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.13.3
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
  Gateway   Distance   Last Update
  1.1.1.1        110   06:26:17
  2.2.2.2        110   06:25:30
  10.1.24.4      110   06:25:30
  Distance: (default is 110)
```

Вывод демонстрирует несколько интересных фактов. Первая выделенная строка повторяет параметры глобальной команды конфигурации `router ospf 1` на маршрутизаторе. Второй выделенный элемент — это идентификатор маршрутизатора R3, обсуждаемый далее в следующем разделе. Третья выделенная строка также повторяет конфигурацию, заданную подкомандой OSPF `network 10.0.0.0 0.255.255.255 area 0`. И наконец, последний выделенный элемент в примере — это заголовок перед списком известных маршрутизаторов OSPF.

Настройка идентификатора маршрутизатора OSPF

Хотя протокол OSPF обладает множеством дополнительных средств, в большинстве корпоративных сетей, использующих протокол OSPF, для каждого маршрутизатора OSPF настраивают идентификатор маршрутизатора. Для правильной работы у всех маршрутизаторов OSPF должен быть *идентификатор маршрутизатора* (Router ID — RID). Изначально в качестве RID маршрутизаторы используют IP-адрес интерфейса. Но большинство сетевых инженеров предпочитают назначать

идентификаторы для каждого маршрутизатора, чтобы вывод команды `show ip ospf neighbor` отображал более осмысленные идентификаторы маршрутизатора.

Для поиска своего идентификатора RID маршрутизатор Cisco использует при перезагрузке и запуске протокола OSPF следующий процесс. При нахождении идентификатора RID на любом из этапов процесс останавливается.

Ключевая тема **Правила установки идентификатора маршрутизатора**

Если подкоманда OSPF `router-id` *rid* введена, то используется ее значение RID.

1. Если на каком-либо петлевом интерфейсе задан IP-адрес и этот интерфейс находится в состоянии `up`, то маршрутизатор выбирает из петлевых интерфейсов самый большой IP-адрес.
2. Маршрутизатор выбирает самый большой IP-адрес из всех остальных интерфейсов, с первым кодом состояния `up`. (Другими словами, интерфейс в состоянии `up/down` будет включен протоколом OSPF в состав кандидатов при выборе идентификатора маршрутизатора.)

Первый и третий критерии должны быть понятны сразу: идентификатор RID либо настраивается, либо выбирается из IP-адресов работающих интерфейсов. Однако концепция *петлевого интерфейса* (loopback interface), упомянутого во втором критерии, в этой книге еще не рассматривалась.

Петлевой интерфейс — это виртуальный интерфейс, допускающий настройку командой `interface loopback` *номер_интерфейса*, где *номер_интерфейса* — целое число. Петлевые интерфейсы всегда находятся в состоянии `up/up`, если они не отключены административно. Например, простая команда конфигурации `interface loopback 0`, сопровождаемая командой `ip address 2.2.2.2 255.255.255.0`, создала бы петлевой интерфейс и присвоила ему IP-адрес. Поскольку петлевые интерфейсы не полагаются ни на какие аппаратные средства, они переходят в состояние `up/up` сразу после запуска IOS, что делает их хорошей основой для выбора идентификаторов RID OSPF.

Пример 17.8 демонстрирует конфигурацию на маршрутизаторах R1 и R2 перед созданием вывода команды `show` в примерах 17.4, 17.5 и 17.6. Маршрутизатор R1 устанавливает идентификатор маршрутизатора, используя прямой метод, а маршрутизатор R2 — петлевой IP-адрес.

Пример 17.8. Примеры установки идентификатора маршрутизатора OSPF

```
! Сначала настройка R1
router ospf 1
  router-id 1.1.1.1
  network 10.1.0.0 0.0.255.255 area 0
  network 10.0.0.0 0.255.255.255 area 0

! Затем настройка R2
!
interface Loopback2
  ip address 2.2.2.2 255.255.255.255
```

Каждый маршрутизатор OSPF выбирает свой идентификатор RID при инициализации протокола OSPF, осуществляемой при запуске маршрутизатора или когда пользователь CLI останавливает и перезапускает процесс OSPF (командой `clear ip ospf process`). Поэтому, если процесс OSPF уже выполняется, но впоследствии конфигурация изменяется так, что это повлияет на RID OSPF, протокол OSPF не изменит идентификатор RID немедленно, а только после следующего перезапуска процесса OSPF.

Пример 17.9 демонстрирует вывод команды `show ip ospf` на маршрутизаторе R1 уже после применения конфигурации из примера 17.8 и перезагрузки маршрутизатора, обеспечившей смену идентификатора маршрутизатора OSPF.

Пример 17.9. Подтверждение текущего идентификатора маршрутизатора OSPF

```
R1# show ip ospf
  Routing Process "ospf 1" with ID 1.1.1.1
! Строки опущены для краткости
```

Дополнительные параметры настройки OSPF

В последних разделах этой главы рассматривается несколько несвязанных дополнительных параметров настройки протокола OSPF, а именно: как сделать интерфейс маршрутизатора пассивным для протокола OSPF и как создать и разослать стандартный маршрут OSPF.

Пассивные интерфейсы OSPF

Как только протокол OSPF будет включен на интерфейсе, маршрутизатор попытается обнаружить соседние маршрутизаторы OSPF и сформировать соседские отношения. Для этого маршрутизатор посылает сообщения Hello OSPF через регулярные интервалы времени (интервалы Hello). Маршрутизатор принимает также входящие сообщения Hello от потенциальных соседей.

Иногда маршрутизатор не должен создавать соседских отношений на интерфейсе. Зачастую на конкретном канале связи нет другого маршрутизатора, поэтому у маршрутизатора нет никакой необходимости продолжить посылать регулярные сообщения Hello OSPF.

Когда маршрутизатору не нужно обнаруживать соседей на каком-либо интерфейсе, у сетевого инженера есть несколько возможностей настройки. Можно не делать ничего, и маршрутизатор продолжит посылать сообщения, растрачивая впустую немного мощности процессора и пропускной способности. Либо инженер может настроить интерфейс как пассивный интерфейс OSPF, указав маршрутизатору сделать следующее.

Действия IOS при пассивном интерфейсе OSPF

Ключевая тема

- Перестает посылать сообщения Hello OSPF на интерфейсе.
- Игнорировать полученные сообщения Hello на интерфейсе.
- Не создает соседских отношений на интерфейсе.

Когда интерфейс становится пассивным, протокол OSPF не создает соседских отношений, но все еще анонсирует подсети, подключенные к данному интерфейсу. Таким образом, конфигурация включает протокол OSPF на интерфейсе (подкомандой

маршрутизатора network), а затем делает интерфейс пассивным (подкомандой маршрутизатора passive-interface).

Существуют две возможности настроить интерфейс как пассивный. Можно добавить в режиме настройки маршрутизатора следующую команду в конфигурацию процесса OSPF:

passive-interface *тип номер*

Либо можно изменить стандартную конфигурацию так, чтобы все интерфейсы стали пассивными, а затем добавить для всех интерфейсов, которые не должны быть пассивными, команду no passive-interface:

passive-interface default
no passive interface *тип номер*

В типичном примере объединенной сети на рис. 17.10 интерфейс LAN маршрутизатора R1 в левой нижней части рисунка настроен как магистральное соединение VLAN. Единственный маршрутизатор, подключенный к обеим VLAN, — это маршрутизатор R1, поэтому он никогда не будет обнаруживать соседей OSPF в этих подсетях. Пример 17.10 демонстрирует две альтернативные конфигурации, которые делают два субинтерфейса LAN пассивными к OSPF.

Пример 17.10. Настройка пассивных интерфейсов на маршрутизаторах R1 и R2 согласно рис. 17.10

```
! Сначала сделать каждый субинтерфейс пассивным непосредственно
router ospf 1
  passive-interface gigabitethernet0/0.11
  passive-interface gigabitethernet0/0.12

! Или изменить значение состояния на пассивное, а некоторые интерфейсы
! сделать активными
router ospf 1
  passive-interface default
  no passive-interface serial0/0/0
  no passive-interface serial0/0/1
```

В реальных объединенных сетях выбор стиля конфигурации сокращает выбор обязательных команд до наименьшего количества. Например, у маршрутизатора с 20 интерфейсами, 18 из которых пассивны для OSPF, будет гораздо меньше команд конфигурации при использовании команды passive-interface default, изменяющей стандартное состояние на пассивное. Если пассивными должны быть только два из тех 20 интерфейсов, то для сокращения конфигурации используйте стандартную настройку, по которой все интерфейсы не пассивны.

Интересно то, что протокол OSPF испытывает нечто вроде проблем при использовании команд show для выяснения, пассивен ли интерфейс. Команда show running-config выводит конфигурацию непосредственно, но если нет возможности перейти в привилегированный режим, чтобы использовать эту команду, то обратите внимание на два следующие факта.

Команда show ip ospf interface brief выводит все интерфейсы, на которых включен протокол OSPF, *включая пассивные интерфейсы.*

Команда `show ip ospf interface` выводит одну строку, указывающую, что интерфейс пассивен.

Пример 17.11 демонстрирует эти две команды на маршрутизаторе R1 при конфигурации из верхней части примера 17.10. Оба субинтерфейса, G0/0.11 и G0/0.12, представлены в выводе команды `show ip ospf interface brief`.

Пример 17.11. Отображение пассивных интерфейсов

```
R1# show ip ospf interface brief
Interface  PID  Area  IP Address/Mask  Cost  State  Nbrs F/C
Gi0/0.12   1    0     10.1.1.129/25    1     DR     0/0
Gi0/0.11   1    0     10.1.1.1/25      1     DR     0/0
Se0/0/0    1    0     10.1.12.1/25     64    P2P    0/0
Se0/0/1    1    0     10.1.13.1/25     64    P2P    0/0

R1# show ip ospf interface g0/0.11
GigabitEthernet0/0.1 is up, line protocol is up
  Internet Address 10.1.1.1/25, Area 0, Attached via Network Statement
  Process ID 1, Router ID 10.1.1.129, Network Type BROADCAST, Cost: 1
  Topology-MTID  Cost  Disabled  Shutdown  Topology Name
       0          1      no        no         Base
  Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 10.1.1.129, Interface address 10.1.1.1
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  No Hellos (Passive interface)
! Строки опущены для краткости
```

Стандартные маршруты OSPF

Как упоминалось в главе 16, маршрутизаторам иногда полезно использовать стандартные маршруты. Там рассматривалась настройка на маршрутизаторе статического стандартного маршрута, используемого только одним маршрутизатором. В данном разделе рассматривается отличная стратегия использования стандартных маршрутов IP, одни из которых маршрутизатор OSPF создает как стандартный маршрут и анонсирует его по протоколу OSPF, а другие маршрутизатор изучает как стандартные маршруты динамически.

Классический пример использования протокола маршрутизации — это анонс стандартного маршрута корпоративного подключения к Интернету. В качестве стратегии сетевой инженер предприятия может использовать следующие подходы.

- Все маршрутизаторы изучают маршруты к конкретным подсетям в компании; у стандартного маршрута нет необходимости в перенаправлении пакетов этим получателям.
- Один маршрутизатор подключается к Интернету, и у него есть стандартный маршрут, указывающий на Интернет.
- Все маршрутизаторы должны динамически изучить стандартный маршрут, используемый для всего трафика Интернета, чтобы все пакеты, предназначенные для областей в Интернете, передавались на один подключенный к Интернету маршрутизатор.

Концепция анонсирования протоколом OSPF стандартного маршрута с конкретной конфигурацией OSPF представлена на рис. 17.11. В данном случае компания под-

ключена к провайдеру ISP через маршрутизатор R1. Этот маршрутизатор использует команду OSPF `default-information originate` (этап 1). В результате, используя протокол OSPF, маршрутизаторы анонсируют стандартный маршрут (этап 2) для дистанционных маршрутизаторов (B1, B2 и B3).

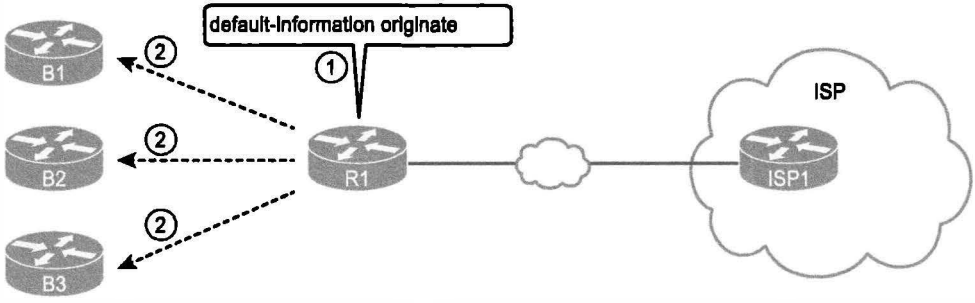


Рис. 17.11. Использование протокола OSPF для создания и рассылки стандартного маршрута

На рис. 17.12 представлены стандартные маршруты, созданные из анонсов OSPF на рис. 17.11. Все три маршрутизатора слева изучили стандартный маршрут OSPF, указывающий на маршрутизатор R1. Сам маршрутизатор R1 также нуждается в стандартном маршруте, указывающем на маршрутизатор провайдера ISP, чтобы он мог перенаправить ему весь трафик, предназначенный для Интернета.

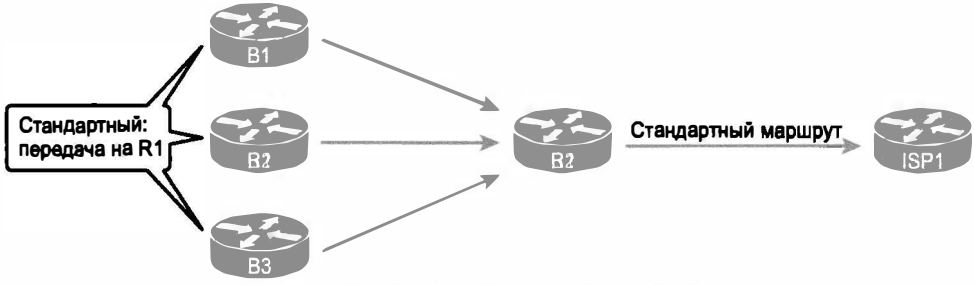


Рис. 17.12. Стандартные маршруты, полученные в результате команды `default-information originate`

И наконец, это средство предоставляет сетевому инженеру контроль над созданием маршрутизатором этого стандартного маршрута. Первоначально маршрутизатор R1 нуждается в стандартном маршруте, определенном либо как статический стандартный маршрут, либо как полученный от провайдера ISP. Затем команда `default-information originate` указывает маршрутизатору R1 анонсировать стандартный маршрут, когда его собственный стандартный маршрут работает, и анонсировать его как отключенный (`down`), когда его собственный стандартный маршрут нарушается.

ВНИМАНИЕ!

Интересно то, что подкоманда маршрутизатора `default-information originate always` указывает маршрутизатору всегда анонсировать стандартный маршрут, независимо от того, работает ли маршрутизатор стандартного маршрута или нет.

Обзор

Резюме

- Маршрутизаторы добавляют маршруты IP в свои таблицы маршрутизации тремя способами: как подключенные маршруты, как статические маршруты и изучает их при помощи протоколов динамической маршрутизации.
- Обобщенно эти термины определяют следующим образом.
  - *Протокол маршрутизации.* Набор сообщений, правил и алгоритмов, используемых маршрутизаторами для общей задачи изучения маршрутов. Этот процесс подразумевает обмен и анализ информации о маршрутизации. Каждый маршрутизатор выбирает наилучший маршрут к каждой подсети (выбор пути), а выбранные оптимальные маршруты помещает в свою таблицу маршрутизации IP. К таким протоколам относятся RIP, EIGRP, OSPF и BGP.
  - *Маршрутизируемый протокол.* Определяет структуру пакета и логику адресации, позволяющие маршрутизаторам перенаправлять (или маршрутизировать) пакеты. Маршрутизаторы перенаправляют пакеты в соответствии с маршрутизируемыми протоколами. К таким протоколам относятся IP версии 4 (IPv4) и версии 6 (IPv6).
- Программное обеспечение Cisco IOS поддерживает несколько протоколов маршрутизации IP, осуществляющих те же общие функции.
  1. Изучать информацию о маршрутах к подсетям IP от других соседних маршрутизаторов.
  2. Анонсировать информацию о маршрутах к подсетям IP другим соседним маршрутизаторам.
  3. Если к подсети возможно несколько маршрутов, то наилучший выбирается на основании метрик.
  4. Если топология сети изменяется (например, откажет канал связи), то некоторые маршруты окажутся неверны и придется выбирать новый оптимальный маршрут. (Этот процесс называется конвергенцией.)
- Термин *конвергенция* относится к процессу, происходящему при изменении топологии сети, т.е. когда отказывает маршрутизатор или канал связи (либо когда они восстанавливают работу). Когда нечто изменяется, могут измениться и наилучшие доступные в сети маршруты. Конвергенция — это процесс, в ходе которого все маршрутизаторы осознают, что произошло некое изменение, анонсирует информацию об изменениях всем остальным маршрутизаторам, а они выбирают для каждой подсети новые оптимальные маршруты. Способность к быстрой конвергенции (без циклов) является одним из важнейших критериев при выборе протокола маршрутизации IP.
- Протоколы маршрутизации IP относятся к одной из двух главных категорий: протоколы маршрутизации внутреннего шлюза (IGP) и протоколы маршрутизации внешнего шлюза (EGP). Как они определяются, описано ниже.

  - *IGP*. Протокол маршрутизации, предназначенный для использования в одиночной автономной системе.
  - *EGP*. Протокол маршрутизации, предназначенный для совместного использования разными автономными системами.
- *Автономная система* (AS) — это сеть под административным контролем одной организации. Например, сеть, созданная и оплаченная некой компанией, будет вероятней всего системой AS, школьная сеть также вероятно является автономной системой.
- Каждой системе AS может быть присвоен номер AS (ASN). Правом присвоения номеров ASN, как и открытых IP-адресов, обладает центр по присвоению адресов Интернета (IANA `www.iana.org`).
- У предприятий есть несколько вариантов при выборе протокола IGP для своей корпоративной сети, но большинство компаний используют протокол OSPF или EIGRP.
- Базовые алгоритмы протокола маршрутизации определяют то, как они выполняют свою задачу. Термин *алгоритм протокола маршрутизации* относится к логике, используемой различными протоколами маршрутизации для изучения всех возможных маршрутов и выбора наилучшего для каждой подсети, а также реакции на изменения в объединенной сети (конвергенции).
- Протоколы маршрутизации IGP используют три основных типа алгоритмов протокола маршрутизации.
  - Дистанционно-векторный (или алгоритм Беллмана–Форда, по имени его создателей).
  - Улучшенный дистанционно-векторный (или "сбалансированный гибридный").
  - Состояния канала.
- Протоколы маршрутизации выбирают наилучший маршрут к подсети на основании самой низкой метрики маршрута. Протокол RIP, например, использует счетчик количества маршрутизаторов (транзитных участков) между текущим маршрутизатором и подсетью назначения. Протокол OSPF подсчитывает цену каждого интерфейса в сквозном маршруте, а также цену на основании ширины полосы пропускания канала связи.
- Административное расстояние — это числовое представление достоверности протокола маршрутизации на одном маршрутизаторе. Чем ниже это значение, тем лучше, тем достоверней протокол маршрутизации.
- Протоколы состояния канала создают маршруты IP в ходе нескольких этапов. Сначала маршрутизаторы совместно выясняют достаточно много информации о сети: ее маршрутизаторы, каналы связи, IP-адреса, информацию о состоянии и т.д. Затем они рассылают эту информацию, чтобы все маршрутизаторы знали ее. На данный момент любой маршрутизатор может вычислять маршруты ко всем подсетям со своей точки зрения.

- Открытый протокол поиска первого кратчайшего маршрута (OSPF) — наиболее популярный протокол маршрутизации IP по состоянию канала — организует информацию о топологии, используя анонсы состояния канала (LSA) и базу данных состояний каналов (LSDB).
- Для получения маршрутов маршрутизаторы осуществляют математические вычисления. К счастью, ни мне ни вам не нужно знать математику столь глубоко! Тем не менее для обработки базы LSDB все протоколы состояния канала используют математический алгоритм поиска первого кратчайшего пути Дейкстры (SPF).
- Для построения маршрутов протокол OSPF использует операции трех основных категорий.
  - *Соседи.* Отношения между двумя маршрутизаторами, соединенными тем же каналом связи. Соседние маршрутизаторы способны обмениваться базами LSDB.
  - *Обмен базами данных.* Процесс пересылки анонсов LSA соседям, чтобы все маршрутизаторы изучили этот анонс.
  - *Добавление наилучших маршрутов.* Процесс вычисления по локальной копии базы LSDB наилучших маршрутов и добавления их в таблицу маршрутизации IPv4 на каждом поддерживающем алгоритм SPF маршрутизаторе.
- Этапы настройки протокола OSPF, а также краткий перечень необходимых команд.

  Этап 1 Перейдите в режим конфигурации определенного процесса OSPF, используя глобальную команду `router ospf` *`идентификатор_процесса`*

  Этап 2 (Необязательно.) Настройка идентификатора маршрутизатора OSPF:

  A. Настройка подкоманды маршрутизатора `router-id` *`значение_идентификатора`*.

  B. Настройка IP-адреса на *петлевом интерфейсе* (loopback interface)

  Этап 3 Настройка одной или нескольких подкоманд маршрутизатора `network` *`ip-адрес шаблон_маски`* `area` *`идентификатор_области`* со всеми соответствующими интерфейсами, добавляемыми к указанным областям

Для правильной работы у всех маршрутизаторов OSPF должен быть идентификатор маршрутизатора (RID). Изначально в качестве RID маршрутизаторы используют IP-адрес интерфейса. Но большинство сетевых инженеров предпочитают назначать идентификаторы для каждого маршрутизатора, чтобы вывод команды `show ip ospf neighbor` отображал более осмысленные идентификаторы маршрутизатора.

- Для поиска своего идентификатора RID маршрутизатор Cisco использует при перезагрузке и запуске протокола OSPF следующий процесс. При нахождении идентификатора RID на любом из этапов процесс останавливается.
- Если подкоманда OSPF `router-id` *`rid`* введена, то используется ее значение RID.

- Если на некоем петлевом интерфейсе задан IP-адрес и этот интерфейс находится в состоянии up, то маршрутизатор выбирает из петлевых интерфейсов самый большой IP-адрес.
- Маршругизатор выбирает самый большой IP-адрес из всех остальных интерфейсов, с первым кодом состояния up. (Другими словами, интерфейс в состоянии up/down будет включен протоколом OSPF в состав кандидатов при выборе идентификатора маршрутизатора.)

Контрольные вопросы

1. Какие из следующих протоколов маршрутизации используют логику состояния канала? (Выберите два ответа).

   А) RIP-1.

   Б) RIP-2.

   В) EIGRP.

   Г) OSPF.

   Д) Integrated IS-IS.

2. Какие из перечисленных ниже протоколов маршрутизации используют метрику, полностью или частично зависящую от полосы пропускания? (Выберите два ответа).

   А) RIP-1.

   Б) RIP-2.

   В) EIGRP.

   Г) OSPF.

3. Какие из следующих внутренних протоколов маршрутизации поддерживают маски VLSM? (Выберите четыре ответа).

   А) RIP-1.

   Б) RIP-2.

   В) EIGRP.

   Г) OSPF.

   Д) Integrated IS-IS.

4. Что из следующего справедливо для маршрутизатора, использующего протокол маршрутизации по состоянию канала для выбора наилучшего маршрута к подсети?

   А) Маршрутизатор находит наилучший маршрут в базе данных состояний каналов.

   Б) Маршрутизатор вычисляет наилучший маршрут, обработав по алгоритму SPF информацию в базе данных состояния каналов.

   В) Маршрутизатор сравнивает метрики, указанные для данной подсети в анонсах, полученных от каждого соседа, и выбирает наилучший маршрут по самой низкой метрике.

   Г) Маршрутизатор использует путь с наименьшим количеством транзитных переходов.

5. Для вычисления оптимального в настоящее время маршрута протокол OSPF применяет некий алгоритм. Какие из следующих терминов описывают этот алгоритм? (Выберите два ответа).

   А) SPF.

   Б) DUAL.

   В) Возможного преемника.

   Г) Дейкстры.

   Д) Здравый смысл.

6. Какая из следующих команд `network`, после команды `router ospf 1`, указывает данному маршрутизатору запустить и использовать протокол OSPF на интерфейсах с IP-адресами 10.1.1.1, 10.1.100.1 и 10.1.120.1?

   А) `network 10.0.0.0 255.0.0.0 area 0`

   Б) `network 10.0.0.0 0.255.255.255 area 0`

   В) `network 10.0.0.1 0.0.0.255 area 0`

   Г) `network 10.0.0.1 0.0.255.255 area 0`

7. Какая из следующих команд `network`, после команды `router ospf 1`, указывает данному маршрутизатору запустить и использовать протокол OSPF на интерфейсах с IP-адресами 10.1.1.1, 10.1.100.1 и 10.1.120.1?

   А) `network 0.0.0.0 255.255.255.255 area 0`

   Б) `network 10.0.0.0 0.255.255.0 area 0`

   В) `network 10.1.1.0 0.x.1x.0 area 0`

   Г) `network 10.1.1.0 255.0.0.0 area 0`

   Д) `network 10.0.0.0 255.0.0.0 area 0`

8. Какие из следующих команд выводят соседей OSPF для последовательного интерфейса 0/0? (Выберите два ответа).

   А) `show ip ospf neighbor`

   Б) `show ip ospf interface brief`

   В) `show ip neighbor`

   Г) `show ip interface`

   Д) `show ip ospf neighbor serial 0/0`

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. Ниже.

Таблица 17.5. Ключевые темы главы 17

Элемент	Описание	Страница
Список	Список основных функций протокола маршрутизации	503
Список	Определения протоколов IGP и EGP	504
Список	Три типа алгоритмов протокола маршрутизации IGP	506

Окончание табл. 17.5

Элемент	Описание	Страница
Табл. 17.1	Сравнение метрик IGP	507
Табл. 17.2	Сравнение внутренних протоколов маршрутизации IP	508
Табл. 17.3	Стандартные административные расстояния	509
Рис. 17.8	Терминология множества областей OSPF	516
Список	Этапы настройки протокола OSPF	517
Список	Шаблон маски OSPF и его значение	519
Список	Правила установки идентификатора маршрутизатора	524
Список	Действия IOS при пассивном интерфейсе OSPF	525

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

конвергенция (convergence), алгоритм поиска кратчайших маршрутов (Shortest Path First — SPF), дистанционно-векторный (distance vector), протокол маршрутизации внутреннего шлюза (Interior Gateway Protocol — IGP), состояние канала (link-state), анонс состояния канала (Link-State Advertisement — LSA), база данных состояний каналов (Link-State Database — LSDB), метрика (metric), маршрутизируемый протокол (routed protocol), протокол маршрутизации (routing protocol), граничный маршрутизатор зоны (Area Border Router — ABR), сосед (neighbor), идентификатор маршрутизатора (Router ID — RID)

Таблицы команд

Хотя и не обязательно заучивать информацию из таблиц данного раздела, в табл. 17.6 приведен список команд конфигурации, а в табл. 17.7 пользовательские команды главы. Команды стоит запомнить, чтобы лучше понять содержимое главы и выполнить задачи по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 17.6. Команды конфигурации главы 17

Команда	Описание
`router ospf` *идентификатор_процесса*	Переводит в режим конфигурации OSPF для указанного процесса
`network` *ip-адрес шаблон_маски* `area` *идентификатор_области*	Подкоманда маршрутизатора, включающая протокол OSPF на интерфейсах, соответствующих комбинации адреса и шаблона, а также устанавливающая область OSPF

Окончание табл. 17.6

Команда	Описание
`auto-cost reference-bandwidth` *номер*	Подкоманда маршрутизатора, задающая делимое в формуле *Ref-BW/Int-BW* вычисления цены на основании ширины полосы пропускания интерфейса
`router-id` *идентификатор*	Подкоманда OSPF, статически устанавливающая идентификатор маршрутизатора
`passive-interface` *тип номер*	Подкоманда OSPF, делающая протокол OSPF пассивным на заданном интерфейсе или субинтерфейсе
`passive-interface default`	Подкоманда OSPF, изменяющая стандартное состояние протокола OSPF для интерфейсов на пассивное вместо активного
`no passive-interface` *тип номер*	Подкоманда OSPF, делающая протокол OSPF активным (не пассивный) на заданном интерфейсе или субинтерфейсе
`default-information originate [always]`	Подкоманда OSPF, требующая создать и анонсировать стандартный маршрут OSPF, пока у маршрутизатора есть некий стандартный маршрут (или всегда анонсировать стандартный маршрут, если задействован параметр `always`)

Таблица 17.7. Пользовательские команды главы 17

Команда	Описание
`show ip route [ospf]`	Выводит список маршрутов в таблице маршрутизации, изученных по протоколу OSPF
`show ip protocols`	Выводит параметры протокола маршрутизации и текущие значения таймеров
`show ip ospf interface [`*тип номер*`]`	Выводит группу сообщений для каждого интерфейса (или только для одного заданного интерфейса) со многими подробностями: область расположения интерфейса, соседи этого интерфейса и интервалы Hello
`show ip ospf interface brief`	Выводит по одной строке для каждого интерфейса с выполняющимся протоколом OSPF, включая пассивные интерфейсы
`show ip ospf neighbor [`*rid_соседа*`]`	Выводит список соседей и их текущее состояние по интерфейсам, а также (дополнительно) подробности о маршрутизаторе, идентификатор которого указан в команде
`show ip ospf database`	Выводит отчет по анонсам LSA в базе LSDB локального маршрутизатора (по одной строке на каждый анонс LSA)
`show ip ospf`	Выводит список фактов о процессе OSPF локального маршрутизатора, в том числе идентификатор маршрутизатора
`show ip protocols`	Выводит группу сообщений для каждого экземпляра каждого протокола маршрутизации, работающего в маршрутизаторе, перечисляя многие параметры конфигурации, стандартные параметры конфигурации, когда они используются, и известные маршрутизаторы OSPF в данной области

Ответы на контрольные вопросы:

1 Г и Д. **2** В и Г. **3** Б, В, Г, и Д. **4** Б. **5** А и Г. **6** Б. **7** А. **8** А и Д.

ГЛАВА 18

Настройка и проверка подключения хостов

В мире TCP/IP слово *хост* (host) означает любое устройство, обладающее IP-адресом: телефон, планшет, компьютер, маршрутизатор, коммутатор или беспроводная точка доступа. Хостами являются даже куда менее очевидные устройства: рекламный видеоэкран на улице, электрический счетчик, использующий для передачи информации о потребленном токе ту же технологию, что и мобильные телефоны, и даже новый автомобиль.

Независимо от типа, любой хост, использующий протокол IPv4, для правильной работы нуждается в четырех параметрах:

- IP-адрес;
- маска подсети;
- адрес стандартного маршрутизатора;
- IP-адрес сервера DNS.

Эта последняя глава данной части завершает обсуждение создания простой сети IPv4 рассмотрением параметров протокола IPv4 на хостах. Начинается глава с описания способов динамического изучения хостами этих четырех параметров при помощи *протокола динамического конфигурирования хостов* (Dynamic Host Configuration Protocol — DHCP). В следующем разделе рассматривается несколько советов по проверке наличия у хоста всех четырех параметров IPv4. И в заключительном разделе описаны три инструмента (ping, traceroute и telnet), позволяющих подтвердить фактическое существование и работоспособность параметров IP.

В этой главе рассматриваются следующие экзаменационные темы

Технологии коммутации сетей LAN

- Проверка состояния сети и работоспособности коммутатора с помощью базовых сетевых утилит ping, telnet и ssh.
- Технологии маршрутизации IP
- Настройка и проверка применения CLI для установки базовой конфигурации маршрутизатора.
  - Команды Cisco IOS для базовой настройки маршрутизатора.
- Проверка конфигурации маршрутизатора и сетевого подключения.
  - Команды Cisco IOS для просмотра базовой информации маршрутизатора и сетевого подключения.
- Службы IP

Настройка и проверка DHCP (маршрутизатор IOS).

- Настройка интерфейса маршрутизатора для использования DHCP.
- Параметры DHCP.
- Исключенные адреса.
- Период резервирования.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Настройка маршрутизаторов на поддержку протокола DHCP

Протокол динамического конфигурирования хоста (DHCP) является одним из наиболее популярных протоколов в сети TCP/IP. Подавляющее большинство хостов в сети TCP/IP — это пользовательские устройства, а подавляющее большинство пользовательских устройств изучает свои параметры IPv4, используя протокол DHCP.

Использование протокола DHCP имеет несколько преимуществ перед настройкой вручную или статическим заданием параметров IPv4. Конфигурация параметров протокола IP для хоста находится на сервере DHCP, а клиент изучает эти параметры, используя сообщения DHCP. В результате конфигурация протокола IP на хосте контролируется инженерно-техническим персоналом, что сокращает пользовательские ошибки. Протокол DHCP допускает присвоение хостам постоянных адресов, но обычно он назначает временные, резервируемые IP-адреса. Благодаря резервированию сервер DHCP может возвращать IP-адреса удаленных из сети устройств, а также лучше использовать доступные адреса.

Протокол DHCP обеспечивает также мобильность. Например, каждый раз, когда пользователь перемещается со своим планшетом в новое место (кафе, квартира, офис), его устройство подключается к беспроводной LAN и использует протокол DHCP для резервирования нового IP-адреса и начала работы в новой сети. Без протокола DHCP пользователь вынужден был бы выяснять информацию о локальной сети и настраивать параметры вручную, при этом ошибки весьма вероятны.

Хотя на пользовательских хостах протокол DHCP работает автоматически, на маршрутизаторах он требует некоторой подготовки и настройки. В некоторых корпоративных сетях конфигурация маршрутизатора на многих его интерфейсах LAN может быть осуществлена одной командой (`ip helper-address` *`ip-адрес_сервера`*), задающей сервер DHCP по его IP-адресу. В других случаях маршрутизатор фактически играет роль сервера DHCP. Независимо от случая, у маршрутизаторов есть некая роль, которую ему приходится играть.

Данный раздел начинается с рассмотрения протокола DHCP, выполняющегося между клиентом DHCP (любым хостом) и сервером DHCP. В следующем разделе рассматриваются минимальные настройки параметров маршрутизатора, используемые отдельным устройством или сервером, являющимся сервером DHCP. Заключительная часть этого раздела посвящена настройке маршрутизатора Cisco как сервера DHCP.

Сообщения протокола DHCP и адреса

Давайте на мгновение задумаемся о роли протокола DHCP для компьютера хоста. Хост действует как клиент DHCP. Как клиент DHCP хост начинает работу без параметров IPv4: без IPv4-адреса, без маски, без IP-адресов стандартного маршрутизатора и сервера DNS. Но у клиента DHCP есть протокол DHCP, и он может использовать его для обнаружения сервера DHCP и передачи ему запроса на предоставление IPv4-адреса.

В процессе предоставления IP-адреса протоколом DHCP между клиентом и сервером передаются следующие четыре сообщения. (Чтобы проще запомнить сообщения — их первые символы складываются в слово DORA).

Четыре сообщения DHCP, используемых при предоставлении нового IP-адреса Ключевая тема

Discover (обнаружить). Передается клиентом DHCP при поиске подходящего сервера DHCP.

Offer (предложение). Передается сервером DHCP как предложение клиенту некоего IP-адреса (и сообщение других параметров).

Request (запрос). Передается клиентом DHCP как запрос серверу на резервирование IPv4-адреса, указанного им в сообщении Offer.

Acknowledgment (подтверждение). Передается сервером DHCP при назначении адреса и информировании о маске, а также IP-адресах стандартного маршрутизатора и сервера DNS.

У клиентов DHCP есть специфическая проблема: у них еще нет IP-адреса, но они должны посылать пакеты IP. Для решения этой проблемы сообщения DHCP используют два специальных IPv4-адреса, позволяющих хосту без IP-адреса посылать и получать сообщения по локальной подсети.

Специальные IPv4-адреса 0.0.0.0 и 255.255.255.255 Ключевая тема

- `0.0.0.0`. Специально зарезервированный IPv4-адрес для использования хостами, у которых еще нет IP-адреса.
- `255.255.255.255`. Адрес, зарезервированный как широковещательный адрес локальной подсети. Посланные на этот адрес пакеты распространяются по локальному каналу связи, но в другие подсети маршрутизаторы их не перенаправляют.

Для демонстрации работы этих адресов на рис. 18.5 приведен пример применения этих IP-адресов при передаче сообщений DHCP между хостом (A) и сервером в той же сети LAN. Хост A, клиент, посылает сообщение Discover с IP-адресом отправителя 0.0.0.0, поскольку у него еще нет собственного IP-адреса. Хост A посылает пакет на адрес получателя 255.255.255.255 — широковещательный адрес сети LAN. Такой пакет доставляется на все хосты в подсети. Клиент полагает, что в локальной подсети есть сервер DHCP. Почему? Пакеты, посланные на адрес 255.255.255.255, поступают только на хосты в локальной подсети; маршрутизатор R1 не будет перенаправлять этот пакет.

ВНИМАНИЕ!

На рис. 18.1 приведен только один применяемый вариант, клиент DHCP вполне может использовать флаг широковещания. В этой главе не рассматриваются все возможные варианты адресов, используемых в соответствии с протоколами DHCP; текущая задача заключается в демонстрации одного из типичных примеров, поэтому здесь есть потребность в специфической функции маршрутизатора — *ретрансляции DHCP* (DHCP Relay). Но обратите внимание, что во всех примерах DHCP этой главы представлены адреса, используемые в случае, если клиент использует флаг широковещания DHCP.

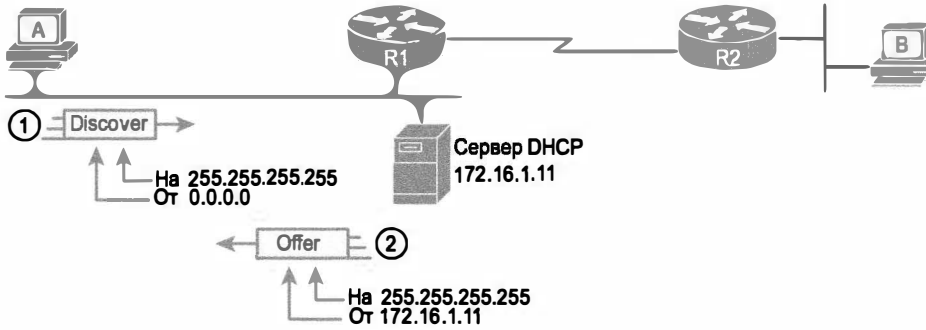


Рис. 18.1. Сообщения DHCP Discover и Offer

Теперь рассмотрим сообщение Offer, возвращаемое сервером DHCP. В качестве IP-адреса получателя сервер снова указывает 255.255.255.255. Почему? Хост все еще не имеет IP-адреса, поэтому сервер не может послать пакет непосредственно на хост A. Поэтому сервер посылает пакет на "все хосты в локальной подсети" (адрес 255.255.255.255), пакет инкапсулируется также в широковещательный фрейм Ethernet. Хост A будет в состоянии получить и обработать сообщение. (Другие хосты получают сообщение, но игнорируют его.)

Сообщения DHCP работают хорошо, когда клиент и сервер DHCP находятся в той же подсети. По завершении обмена этими четырьмя сообщениями клиент DHCP обладает IP-адресом и другими параметрами IPv4, необходимыми для передачи одноадресатных пакетов IP.

Поддержка протокола DHCP для дистанционных подсетей при помощи ретрансляции DHCP

Проектируя применение протокола DHCP, сетевой инженер должен сделать выбор: помещать ли сервер DHCP в каждую подсеть LAN или расположить на центральной площадке? При наличии сервера DHCP в каждой подсети протокол работает, как на рис. 18.1, а маршрутизатор может полностью игнорировать протокол DHCP. Но при централизованном сервере DHCP большинство его клиентов находится в подсетях отличных от той, где расположен сервер DHCP. Исходя из описанного до сих пор, сообщение DHCP никогда не достигало бы сервера DHCP, поскольку маршрутизаторы не перенаправляют пакеты IPv4, посланные на IP-адрес 255.255.255.255.

Большинство корпоративных сетей используют несколько серверов DHCP на централизованной площадке, обеспечивающих услуги протокола DHCP для всех дистанционных подсетей. Маршрутизаторы так или иначе должны перенаправлять эти сообщения DHCP между клиентами и сервером DHCP. Для этого маршрутизаторы, соединенные с дистанционными подсетями LAN, нуждаются в подкоманде интерфейса `ip helper-address` *`ip-адрес_сервера`*.

Подкоманда `ip helper-address` *`ip-адрес_сервера`* указывает маршрутизатору сделать следующее для сообщений, поступающих на интерфейс от клиента DHCP.

Четыре логических этапа, обусловленных командой `ip helper-address`

Ключевая тема

1. Отследить входящие сообщения DHCP с IP-адресом получателя 255.255.255.255.
2. Изменить IP-адрес отправителя пакета на IP-адрес исходящего интерфейса маршрутизатора.
3. Изменить IP-адрес получателя пакета на адрес сервера DHCP (как задано командой `ip helper-address`).
4. Перенаправить пакет на сервер DHCP.

Эта команда меняет правило "не перенаправлять пакеты, посланные по адресу 255.255.255.255", на "изменять IP-адреса получателя". Как только получатель сможет распознать IP-адрес сервера DHCP, сеть сможет перенаправить пакет на сервер.

ВНИМАНИЕ!

Этот подход, подразумевающий ретрансляцию сообщения DHCP за счет изменения IP-адресов в заголовке пакета, называется *ретрансляцией DHCP* (DHCP relay).

Пример процесса приведен на рис. 18.2. Находящийся слева хост A является клиентом DHCP. Сервер DHCP (172.16.2.11) находится справа. На интерфейсе G0/0 маршрутизатора R1 настроена команда `ip helper-address 172.16.2.11`. На этапе 1 маршрутизатор R1 обнаруживает входящий пакет DHCP, предназначенный для адреса 255.255.255.255. Этап 2 демонстрирует результат изменения IP-адресов отправителя и получателя — маршрутизатор R1 перенаправляет пакет.

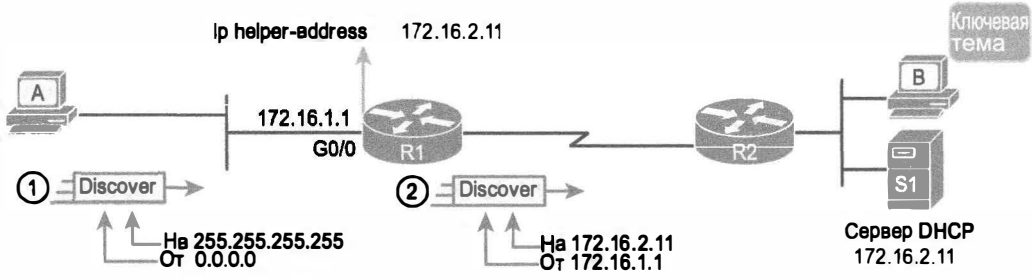


Рис. 18.2. Что команда `ip helper-address` изменяет в сообщении DHCP Discover (эффект вспомогательного IP-адреса)

Подобный процесс маршрутизатор использует для возвращения сообщения DHCP от сервера. Сначала при возвращении пакета с сервера DHCP его IP-адреса отправителя и получателя просто меняются местами относительно пакета, полученного от маршрутизатора (агент пересылки). Например, на рис. 18.2 сообщение Discover имеет IP-адрес отправителя 172.16.1.1, и возвращаемое сервером сообщение Offer имеет IP-адрес получателя 172.16.1.1.

Когда маршрутизатор получает сообщение DHCP, адресованное одному из собственных IP-адресов маршрутизатора, он понимает, что пакет может участвовать в процессе ретрансляции DHCP. Если это так, то агент пересылки DHCP (маршрутизатор R1) должен изменить IP-адрес получателя так, чтобы реальный клиент DHCP (хост A, у которого еще нет IP-адреса) мог получить и обработать пакет. На рис. 18.3

приведен пример этого процесса, когда маршрутизатор R1 получает сообщение DHCP Offer, посланное на его собственный адрес 172.16.1.1. Маршрутизатор R1 изменяет адрес получателя пакета на 255.255.255.255 и передает его на порт G0/0, зная, что все хосты (включая клиента DHCP A) получат это сообщение.

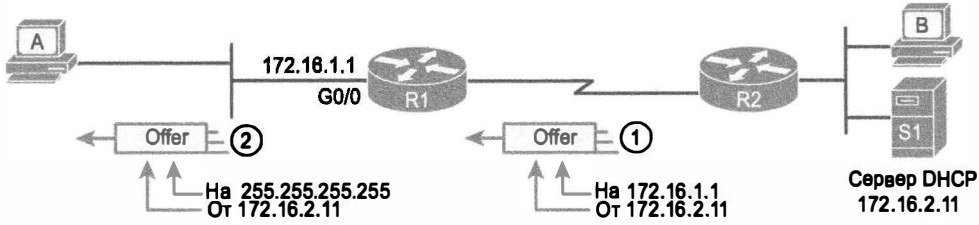


Рис. 18.3. Вспомогательный IP-адрес для сообщения Offer, возвращаемого сервером DHCP

Многие корпоративные сети используют централизованный сервер DHCP, поэтому стандартная конфигурация маршрутизатора включает команду `ip helper-address` для каждого интерфейса и субинтерфейса LAN. При такой стандартной конфигурации пользовательские хосты в любой локальной сети на интерфейсе маршрутизатора всегда могут получить доступ к серверу DHCP и получить резервируемый IP-адрес.

Информация, хранимая на сервере DHCP

Можно подумать, что сервер DHCP похож на громадный системный блок, установленный в большой закрытой серверной комнате с мощным кондиционером для охлаждения воздуха. Однако на самом деле большинство серверов — это программное обеспечение, выполняющееся под некой серверной операционной системой. Сервер DHCP может быть частью программного обеспечения, загруженного бесплатно и установленного на старом компьютере. Но поскольку сервер должен быть доступен постоянно, для поддержки новых клиентов DHCP большинство компаний устанавливает это программное обеспечение на очень стабильный и доступный системный сервер, однако служба DHCP все равно остается программным обеспечением.

Чтобы быть всегда готовым ответить клиенту DHCP и снабдить его IPv4-адресом наряду с другой информацией, сервер DHCP (программное обеспечение) сам нуждается в информации. Обычно серверы DHCP организуют такие параметры IPv4 по подсетям, поскольку информация, сообщаемая клиенту сервером, одинакова для всех хостов в той же подсети. Например, правила IP-адресации гласят, что все хосты в той же подсети должны использовать ту же маску.

Ниже приведены типы параметров, которые должны быть известны серверу DHCP для поддержки клиентов DHCP.

Идентификатор подсети и маска. Сервер DHCP может использовать эту информацию для вычисления всех адресов в подсети. Обычно сервер способен предоставить любой допустимый адрес в подсети, кроме зарезервированных и исключенных. (Серверу DHCP известно, что нельзя предоставлять идентификатор подсети и широковещательный адрес подсети.)

Зарезервированные (исключенные) адреса. Серверу должны быть известны *не предоставляемые* адреса подсети. Этот список позволяет резервировать некоторые адреса от присвоения как IP-адреса, присвоенные статически. Например, статически присваивается большинство IP-адресов маршрутизаторов и коммутаторов, серверов и многих других устройств, кроме пользовательских. Как правило, инженеры используют то же соглашение для всех подсетей, резервируя либо самые младшие IP-адреса во всех подсетях, либо самые старшие.

IP-адрес стандартного маршрутизатора. IP-адрес маршрутизатора данной подсети.

IP-адрес сервера DNS. Список IP-адресов серверов DNS.

На рис. 18.4 приведена концепция, лежащая в основе предварительной конфигурации на сервере DHCP для двух подсетей на базе LAN 172.16.1.0/24 и 172.16.2.0/24. Сервер DHCP находится справа. Для каждой подсети сервер определяет все элементы списка. В данном случае для статических адресов конфигурация резервирует самые младшие IP-адреса в подсети.

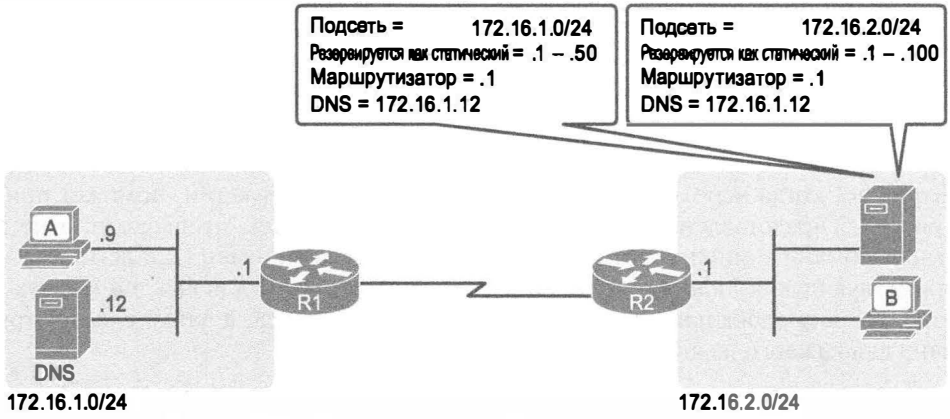


Рис. 18.4. Предварительная конфигурация на сервере DHCP

В конфигурации в наличии могут быть и другие параметры. Например, может быть установлен срок предоставления (резервирования) IP-адреса. Сервер предоставляет адрес на какое-то время (обычно на несколько дней), а затем клиент может запросить продление резервирования. Если клиент его не продлит, сервер может освободить IP-адрес и отложить его в пул доступных IP-адресов. Максимальное время резервирования устанавливает конфигурация сервера.

Настройка и проверка сервера DHCP на маршрутизаторах

Быстрый поиск в Google по ключевым словам "DHCP server products" позволяет найти множество компаний, предоставляющих программное обеспечение сервера DHCP. Маршрутизаторы Cisco (и некоторые коммутаторы Cisco) после небольшой дополнительной настройки также могут выступать в качестве сервера DHCP. В данном разделе описаны настройка и проверка работы протокола DHCP на маршрутизаторе Cisco.

Настройка сервера DHCP

Итак, вы уже ознакомились с видами информации, настраиваемой на сервере. Сервер DHCP на устройстве Cisco не является исключением. Большинство параметров конфигурации сгруппировано по областям (по одной на подсеть), называемым *пулом DHCP* (DHCP pool). Единственной командой DHCP, находящейся вне пула, является команда, определяющая список адресов, исключенных из набора предоставляемых.

Ниже представлены этапы настройки сервера DHCP на базе Cisco IOS.

Ключевая тема

Этапы настройки сервера DHCP на базе Cisco IOS

Этап 1 Исключить адреса из набора предоставляемых сервером DHCP: `ip dhcp excluded-address` *первый последний*

Этап 2 Создать пул DHCP и перейти в режим конфигурации пула: `ip dhcp pool` *имя*

A. Определить подсеть, поддерживаемую сервером DHCP: `network` *идентификатор_подсети маска* или `network` *идентификатор_подсети длина_префикса*

B. Определить IP-адрес (адреса) стандартного маршрутизатора в данной подсети: `default-router` *адрес1 адрес2...*

C. Определить список IP-адресов сервера DNS: `dns-server` *адрес1 адрес2...*

D. Определить продолжительность предоставления в днях, часах и минутах: `lease` *дней часов минут*

E. Определить имя домена DNS: `domain-name` *имя*

Конечно, когда необходимо так много команд конфигурации, поможет пример. На рис. 18.5 представлена конфигурация с использованием псевдокода, а не конкретных команд конфигурации. (Приведенный ниже пример 18.1 демонстрирует соответствующую конфигурацию.) Обратите внимание, что для каждой из двух подсетей LAN есть глобальная команда, исключающая адреса, а затем следует группа команд для каждого из двух пулов DHCP.

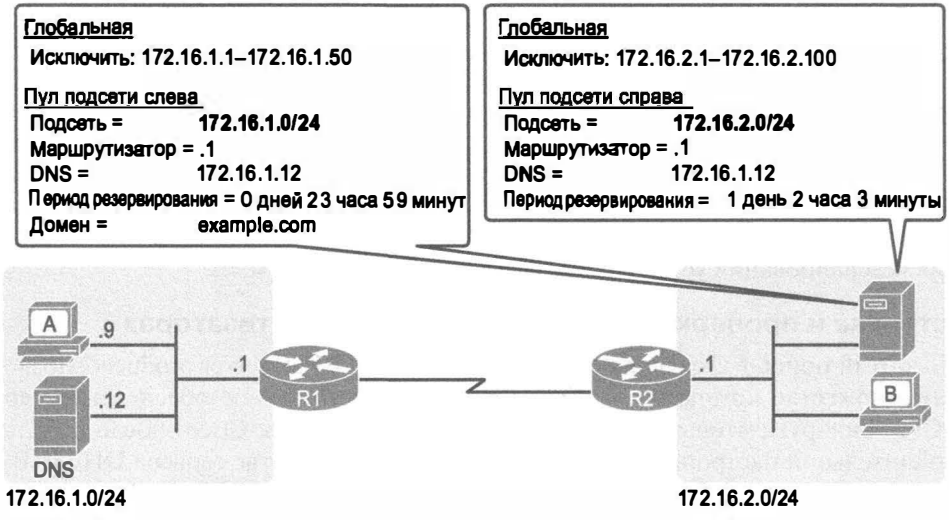


Рис. 18.5. Псевдокод конфигурации сервера DHCP

Пример 18.1. Маршрутизатор R2 как сервер DHCP согласно концепциям, приведенным на рис. 18.5

```
ip dhcp excluded-address 172.16.1.1 172.16.1.50
ip dhcp excluded-address 172.16.2.1 172.16.2.100
!
ip dhcp pool subnet-left
  network 172.16.1.0 255.255.255.0
  dns-server 172.16.1.12
  default-router 172.16.1.1
  lease 0 23 59
  domain-name example.com
!
ip dhcp pool subnet-right
  network 172.16.2.0 /24
  dns-server 172.16.1.12
  default-router 172.16.2.1
  lease 1 2 3
```

Сосредоточимся на подсети 172.16.1.0/24: она настроена как *пул подсети слева* (`pool subnet-left`). Идентификатор подсети и маска соответствуют таковым, выбранным для этой подсети. Далее, глобальная команда `ip dhcp excluded-address` чуть выше определяет диапазон адресов от 172.16.1.1 до 172.16.1.50 как не предоставляемых данным сервером DHCP. Сервер DHCP автоматически исключит идентификатор подсети (172.16.1.0), поэтому он будет предоставлять IP-адреса начиная с адреса .51.

И наконец, обратите внимание на то, что настройка маршрутизатора как сервера DHCP не исключает необходимости в команде `ip helper-address`. Если клиенты DHCP еще существуют в локальных сетях без сервера DHCP, то соединенные с ними маршрутизаторы нуждаются в команде `ip helper-address`. Например, на рис. 18.5 маршрутизатор R1 все еще нуждается в команде `ip helper-address` на интерфейсе LAN.

Проверка сервера DHCP

У сервера DHCP IOS есть несколько других команд `show`, которые приведены ниже.

- `show ip dhcp binding`. Выводит информацию обо всех IP-адресах, предоставленных клиентам в настоящее время.
- `show ip dhcp pool [`*имяпула*`]`. Выводит заданный диапазон IP-адресов, а также статистику по количеству предоставленных в настоящее время адресов и максимальное количество резервируемых адресов по каждому пулу.
- `show ip dhcp server statistics`. Выводит статистику сервера DHCP.

Пример 18.2 демонстрирует типичный вывод двух из этих команд на основании конфигурации рис. 18.5 и примера 18.1. В данном случае сервер DHCP предоставил один IP-адрес от каждого пула, один для хоста A и один для хоста B, как свидетельствуют выделенные строки вывода.

Пример 18.2. Проверка текущего состояния сервера DHCP на базе маршрутизатора

```
R2# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address   Client-ID/            Lease expiration      Type
             Hardware address/
             User name
172.16.1.51  0063.6973.636f.2d30.  Oct 12 2012 02:56 AM  Automatic
             3230.302e.3131.3131.
             2e31.3131.312d.4661.
             302f.30
172.16.2.101 0063.6973.636f.2d30.  Oct 12 2012 04:59 AM  Automatic
             3230.302e.3232.3232.
             2e32.3232.322d.4769.
             302f.30

R2# show ip dhcp pool subnet-right
Pool subnet-right :
Utilization mark (high/low)    : 100 / 0
Subnet size (first/next)       : 0 / 0
Total addresses                : 254
Leased addresses               : 1
Pending event                  : none
1 subnet is currently in the pool :
Current index         IP address range            Leased addresses
172.16.2.102          172.16.2.1 - 172.16.2.254   1
```

Вывод в примере 18.2 не отображает исключенные адреса, но их влияние заметно. Предоставленные клиентам адреса завершаются адресом .51 (хост A, подсеть 172.16.1.0) и адресом .101 (хост B, подсеть 172.16.2.0). Это доказывает, что сервер действительно исключил адреса, как показано в конфигурации примера 18.1. Сервер избежал адресов от .1 до .50 в подсети 172.16.1.0 и адресов от .1 до .100 в подсети 172.16.2.101.

ВНИМАНИЕ!

Сервер DHCP хранит информацию о состоянии каждого получившего адрес клиента DHCP. В частности, он помнит идентификатор клиента DHCP и зарезервированный им IP-адрес. В результате сервер DHCP Ipv4 можно считать сервером DHCP с фиксацией состояния. Это описание будет полезно при чтении главы 28, посвященной протоколу DHCP для Ipv6.

Обнаружение конфликтов предоставляемых и используемых адресов

Сервер DHCP на базе операционной системы Cisco IOS ищет также потенциальные конфликты между предоставляемыми адресами и адресами, настроенными статически. Хотя в конфигурации сервера DHCP непосредственно перечислены адреса в пуле, а также адреса, исключенные из пула, хосты вполне могут статически задать адреса из диапазона в пуле адресов DHCP. Другими словами, никакие протоколы не запрещают хостам статически задать и использовать IP-адреса из диапазона адресов, используемых сервером DHCP.

Зная, что некий хост, возможно, статически настроил адрес из диапазона адресов пула DHCP, и серверы, и клиенты DHCP должны попытаться обнаружить подобные проблемы, прежде чем клиент начнет использовать полученный адрес и вступит в конфликт адресов.

Серверы DHCP обнаруживают конфликты при помощи *эхо-запросов* (ping). Прежде чем предоставить новый IP-адрес клиенту, сервер DHCP отправляет эхо-запрос на этот адрес. Если сервер получает ответ, то он узнает, что некий другой хост уже использует этот адрес и возможен конфликт. Сервер отмечает этот адрес как конфликтный и, не предоставляя его, переходит к следующему адресу в пуле.

Клиент DHCP способен обнаруживать также конфликты, используя протокол ARP вместо эхо-запросов. В этом случае, получив от сервера DHCP предложение использовать некий IP-адрес, клиент DHCP посылает запрос ARP по этому адресу. Если другой хост отвечает, клиент DHCP обнаруживает конфликт.

Пример 18.3 демонстрирует вывод сервера DHCP на базе маршрутизатора R2 после того, как хост B обнаружил конфликт, используя протокол ARP. Внутренне хост B использовал протокол DHCP для запроса на резервирование. Процесс протекал нормально, пока хост B не использовал протокол ARP и не выяснил, что некое другое устройство уже использует адрес 172.16.2.102. В результате хост B отослал назад на сервер сообщение DHCP с отказом использовать адрес 172.16.2.102. Пример демонстрирует регистрационное сообщение маршрутизатора, свидетельствующее, что хост B обнаружил конфликт, а команда show отображает все адреса, находящиеся в конфликте.

Пример 18.3. Отображение информации о конфликтах DHCP

```
*Oct 16 19:28:59.220: %DHCPD-4-DECLINE_CONFLICT: DHCP address conflict:
client
0063.6973.636f.2d30.3230.302e.3034.3034.2e30.3430.342d.4769.302f.30
declined 172.16.2.102.

R2# show ip dhcp conflict
IP address      Detection method    Detection time          VRF
172.16.2.102    Gratuitous ARP      Oct 16 2012 07:28 PM
```

Команда show ip dhcp conflict выводит также метод, использованный сервером для обнаружения конфликта и добавления каждого адреса в список конфликтных. Это может быть либо протокол ARP (на стороне клиента), либо эхо-запрос (со стороны сервера). Сервер не будет предоставлять эти конфликтные адреса будущим клиентам, пока инженер не использует команду clear ip dhcp conflict для очистки списка.

Проверка параметров хоста Ipv4

Одни хосты используют для изучения своих параметров Ipv4 протокол DHCP, другие устанавливают их вручную, третьи фактически позволяют установить некоторые параметры вручную, а некоторые — изучить при помощи протокола DHCP.

Независимо от того, как именно конкретный хост создает свою конфигурацию Ipv4, он будет либо работать, либо иметь проблемы. В случае проблем кто-то должен быть готов (и быть в состоянии) решить проблему. Для хостов это означает, что некто должен быть способен найти параметры протокола Ipv4, проверить их правильность и найти причины неработоспособности, связанные с параметрами IP хоста.

В этом коротком разделе лишь поверхностно затрагивается проверка параметров протокола Ipv4 на хостах. Здесь представлены все параметры и некоторые из команд хоста, обычно используемых для проверки каждого параметра, а также даны некоторые советы по проверке работоспособности. Обратите внимание: экзамены CCNA и ICND2 уделяют особое внимание поиску и устранению неисправностей.

Настройка IP-адреса и маски

Почти каждая операционная система в мире (с каждым днем появляются все новые OS) имеет довольно простое и доступное окно, в котором перечислено большинство, если не все, параметров протокола Ipv4. Например, на рис. 18.6 представлено окно Network (Сеть) операционной системы (в данном случае Mac OS X) пользовательского хоста со всеми параметрами протокола Ipv4. Данный конкретный пример демонстрирует более четырех параметров: IP-адрес, маску подсети, адреса маршрутизатора и сервера DNS.

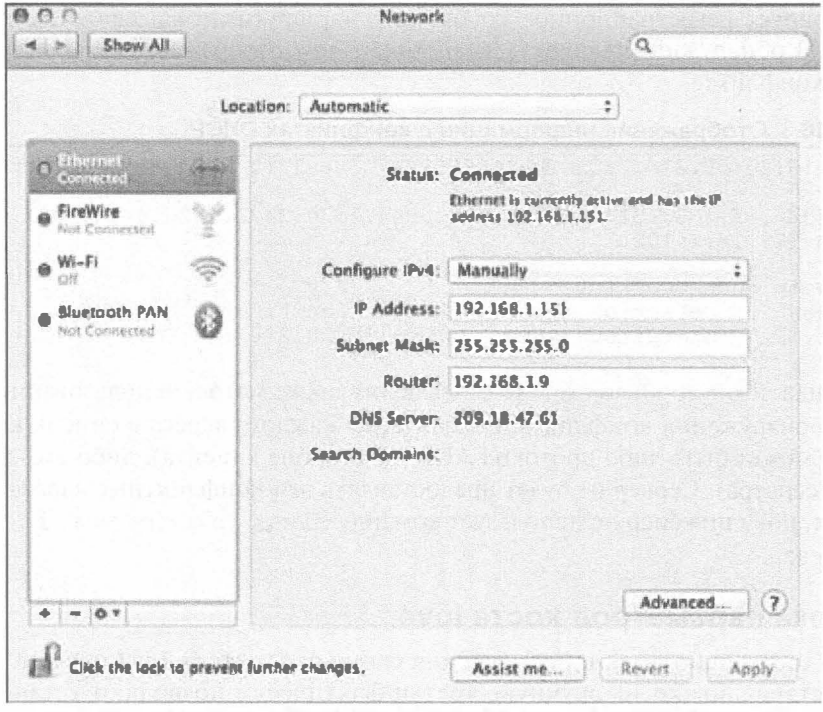


Рис. 18.6. Параметры IP-адреса, маска подсети, адрес стандартного маршрутизатора и адрес сервера DNS в операционной системе Mac OS X

Но кроме всех окон и *графического интерфейса пользователя* (Graphical User Interface — GUI), у большинства операционных систем есть множество сетевых команд, доступных из приглашения командной строки. Интересно то, что некоторые из команд одинаковы в разных операционных системах, даже в разных версиях Microsoft Windows и других OS.

Например, для проверки IP-адреса, маски, стандартного маршрутизатора и других параметров обычно используется команда `ipconfig` (на OS Windows) или `ifconfig` (на Linux и Mac OS). У обеих команд есть несколько параметров, которые можно просмотреть, добавив в конце параметр `-?`. Пример 18.4 демонстрирует вывод для OS Windows.

Пример 18.4. Команда `ipconfig /all` (OS Windows)

```
C:\DOCUME1\OWNER> ipconfig /all
Windows IP Configuration

Ethernet adapter Wireless Network Connection 3:

  Connection-specific DNS Suffix .  : Belkin
  Description . . . . . . . . . . . : Linksys WUSB600N Dual-Band Wireless-N
USB Network Adapter
  Physical Address. . . . . . . . . : 00-1E-E5-D8-CB-E4
  Dhcp Enabled. . . . . . . . . . . : Yes
  Autoconfiguration Enabled . . . . : Yes
  IP Address. . . . . . . . . . . . : 192.168.2.13
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . : 192.168.2.1
  DHCP Server . . . . . . . . . . . : 192.168.2.1
  DNS Servers . . . . . . . . . . . : 192.168.2.1
  Lease Obtained. . . . . . . . . . : Wednesday, October 10, 2012 3:25:00AM
  Lease Expires . . . . . . . . . . : Monday, January 18, 2013 11:14:07 PM
```

Преобразование имен при помощи сервера DNS

Система доменных имен (Domain Name System — DNS) — это и протокол, и международная система серверов, использующая протокол DNS. Будучи очень важным (возможно, одним из самых важных протоколов в мире TCP/IP), протокол DNS не требует никакого внимания от маршрутизаторов и коммутаторов, расположенных между пользовательскими устройствами и серверами DNS. В данном разделе будет показано несколько связанных с DNS команд маршрутизаторов, которые вполне могут пригодиться.

В одной компании вполне может использоваться несколько избыточных серверов DNS, каждый из которых способен преобразовать любые имена для любых хостов в компании. На рис. 18.7 приведен пример одной компании с клиентом (*слева*) и сервером DNS (*сверху*). На этапе 1 представлен запрос DNS на преобразование имени "Server1" в соответствующий ему IP-адрес. Сервер DNS возвращает ответ DNS, содержащий IP-адрес. И наконец, на этапе 3 клиент может послать пакет на адрес 10.1.2.3, соответствующий серверу Server1.

Теперь сосредоточимся на части "На:" этих трех сообщений. У каждого пакета есть заранее известный одноадресатный адрес получателя. Маршрутизаторы в сети TCP/IP просто перенаправляют эти пакеты. Для этого не нужна никакая специальная настройка, никакая функция или такая команда, как `ip helper-address`, используемая для протокола DHCP. Короче говоря, для поддержки протокола DNS на маршрутизаторах и коммутаторах, расположенных между хостом и серверами DNS, не нужно никаких дополнительных действий по настройке.

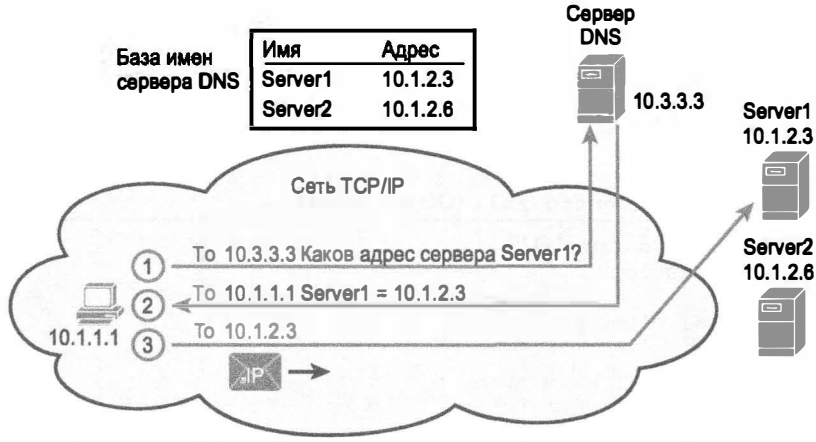


Рис. 18.7. Хост преобразует имя в IP-адрес перед передачей пакета на Server1

Выясняя проблемы с хостами, можно и нужно проверить параметры DNS и убедиться в доступности сервера DNS. В то же время пользователь на хосте может сделать попытку использовать DNS. Например, так, как описано ниже.

- Открыть веб-браузер и ввести имя веб-сервера. Сервер DNS преобразует имя, расположенное между символами // и первым символом /.
- Использовать такую команду, как `nslookup` *`имя_хоста`*, поддерживаемую на большинстве OS. Она посылает запрос на сервер DNS и отображает результаты.

Пример 18.5 демонстрирует команду `nslookup`, подтверждающую, что сервером DNS хоста является 209.18.47.61, а конец вывода свидетельствует о том, что запрос DNS сработал.

Пример 18.5. Команда `nslookup` (Mac OS)

```
Wendell-Odoms-iMac: wendellodom$ nslookup www.certskills.com
Server:                 209.18.47.61
Address:                209.18.47.61#53

Non-authoritative answer:
www.certskills.com        canonical name = certskills.com.
Name: certskills.com
Address: 173.227.251.150
```

Как можно заметить, у маршрутизаторов и коммутаторов действительно немного параметров связано с DNS. Однако параметры DNS позволяют маршрутизатору или коммутатору действовать как преобразователь DNS (клиент). Таким образом, маршрутизатор или коммутатор будет использовать сообщения DNS для запроса у сервера DNS преобразования имени в его IP-адрес. Ниже приведены команды настройки маршрутизаторов и коммутаторов на преобразование имен хостов в их адреса (все команды глобальные).

- `ip name-server` *`IP-адрес_сервера`*... . Позволяет задать до шести IP-адресов серверов DNS.

- `ip host` *имя адрес*. Статически задает одно имя и соответствующий IP-адрес на данном маршрутизаторе или коммутаторе. Локальный маршрутизатор (коммутатор) будет использовать только этот IP-адрес, если команда относится к имени.
- `no ip domain-lookup`. Отключает функцию преобразователя DNS, чтобы маршрутизатор или коммутатор не попытался запрашивать сервер DNS на преобразование имен. (Изначально заданная команда `ip domain-lookup` позволяет маршрутизатору использовать сервер DNS.)

Стандартные маршрутизаторы

Как упоминалось в главе 16, логика перенаправления хоста Ipv4 сводится к простому выбору из двух частей. Пакеты, предназначенные хосту в той же подсети, локальный хост посылает непосредственно, игнорируя любые маршрутизаторы. Пакеты, предназначенные хосту в другой подсети, локальный хост посылает на свой стандартный шлюз (также известный как стандартный маршрутизатор), который и перенаправляет пакет далее.

Интересно, что при передаче между хостом локальной сети и его стандартным маршрутизатором может произойти несколько простых ошибок. Чтобы стандартный маршрутизатор хоста LAN работал правильно, для него должно быть истинно следующее.

Последовательность проверки соответствия параметров хоста Ipv4 параметрам стандартного маршрутизатора Ipv4

Ключевая тема

- Канал связи хоста с локальной сетью и канал связи стандартного маршрутизатора с локальной сетью должны находиться в той же сети VLAN.
- IP-адреса хоста и стандартного маршрутизатора должны принадлежать той же подсети.
- Настроенный на хосте адрес стандартного маршрутизатора должен совпадать с IP-адресом, настроенным на маршрутизаторе. (Другими словами, если хост утверждает, что его стандартный маршрутизатор 10.1.1.1, удостоверьтесь, что IP-адрес интерфейса маршрутизатора не 10.1.1.2.)
- Коммутаторы LAN не должны отбрасывать фреймы из-за настроек защиты порта.

Все упомянутые выше параметры могут не совпадать на хосте и стандартном маршрутизаторе. Параметры на маршрутизаторе можно проверить обычными командами CLI: `show interfaces`, `show ip interface brief`, `show protocols` и `show running-config`. Чтобы проверить параметры VLAN на коммутаторе, используются команды `show interfaces status`, `show vlan` и `show interfaces switchport`.

Методы проверки параметров стандартного маршрутизатора на хосте, конечно, зависят от конкретной OS. Используя GUI, можно просто просмотреть параметры стандартного маршрутизатора. Однако общая для большинства пользовательских OS хостов команда `netstat -rn` указывает стандартный шлюз как маршрут для получателя 0.0.0.0. Пример 18.6 демонстрирует вывод команды `netstat-rn` на

компьютере, работающем под OS Windows, с выделенным параметром стандартного маршрутизатора.

Пример 18.6. Команда `netstat -rn` (OS Windows)

```
C:\DOCUME1\OWNER> netstat -rn
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 11 2f 16 c4 7a ...... NVIDIA nForce Networking Controller -
Packet Scheduler
Miniport
0x3 ...00 1e e5 d8 cb e4 ...... Linksys WUSB600N Dual-Band Wireless-N USB
Network Adapter
- Packet Scheduler Miniport
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface Metric
          0.0.0.0          0.0.0.0    192.168.2.1   192.168.2.13  25
        127.0.0.0        255.0.0.0      127.0.0.1     127.0.0.1  1
    169.254.0.0      255.255.0.0  192.168.2.13  192.168.2.13  20
    192.168.2.0    255.255.255.0  192.168.2.13  192.168.2.13  25
   192.168.2.13  255.255.255.255    127.0.0.1     127.0.0.1  25
  192.168.2.255  255.255.255.255  192.168.2.13  192.168.2.13  25
      224.0.0.0        240.0.0.0  192.168.2.13  192.168.2.13  25
255.255.255.255  255.255.255.255  192.168.2.13             2   1
255.255.255.255  255.255.255.255  192.168.2.13  192.168.2.13   1
Default Gateway: 192.168.2.1
===========================================================================
```

Еще один хороший подход проверки стандартного маршрутизатора — это проверка работы протокола ARP на стандартном маршрутизаторе. Например, хост A на рис. 18.8 посылает пакеты на хост D непосредственно, поскольку он расположен в той же подсети. Таким образом, хосту A нужна запись ARP для хоста D. Аналогично, прежде чем послать пакет на сервер B, находящийся в другой подсети, хост A нуждается в записи ARP для MAC-адреса маршрутизатора R1.

Рис. 18.8. IP- и MAC-адрес хоста в следующих двух примерах ARP

Команда `arp -a` — это еще одна популярная команда, имеющаяся у многих пользовательских OS. Она выводит таблицу ARP хоста. Пример 18.7 демонстрирует таблицу ARP хоста A после успешной передачи им пакетов на сервер B и хост D. Обратите внимание, что IP-адрес сервера B 172.16.2.9 не указан, поскольку таблица ARP содержит IP-адреса хостов собственной подсети и не дистанционных подсетей.

Пример 18.7. Таблица ARP на хосте A (OS Windows)

```
C:\Users\wodom> arp -a

Interface: 172.16.1.9 --- 0xa
  Internet Address      Physical Address      Type
  172.16.1.1            02-00-01-01-01-01     dynamic
  172.16.1.8            00-50-56-e5-d4-72     dynamic
```

Маршрутизаторы должны также хранить таблицу ARP, чтобы иметь возможность инкапсулировать пакеты IP во фреймы LAN. Пример 18.8 демонстрирует вывод команды `show arp` на маршрутизаторе R1, отображая запись для хоста (172.16.1.9) и для самого маршрутизатора (172.16.1.1). (Обычно хосты не хранят собственные IP-адреса в своем кэше ARP, но маршрутизаторы Cisco поступают именно так.)

Пример 18.8. Таблица ARP на маршрутизаторе R1

```
R1# show arp
Protocol Address      Age (min) Hardware Addr   Type  Interface
Internet 172.16.1.1         -   0200.0101.0101  ARPA  GigabitEthernet0/0
Internet 172.16.1.9         2   0200.1111.1111  ARPA  GigabitEthernet0/0
```

Проверка соединения при помощи команд `ping`, `traceroute` и `telnet`

Несмотря на то что проверка параметров Ipv4 на хосте очень важна, окончательная, истинная проверка заключается в проверке возможности приложений хоста общаться так, как должно. Может ли пользователь открыть веб-браузер и просмотреть веб-сайты? Работает ли электронная почта? Все ли приложения могут подключаться к Интернету?

В этом разделе рассматривается несколько средств проверки подключения, отвечающих на простой, но важный вопрос: может ли хост передавать пакеты на другой хост и принимать ответные пакеты? Пользовательские приложения не могут работать, пока эти два хоста не могут обмениваться пакетами. Если инструментальные средства доказывают, что пакеты Ipv4 действительно могут передаваться между двумя хостами, но приложение все равно не работает, то процесс поиска неисправности следует свести к поиску проблем самого приложения. Но если хосты не могут передавать пакеты друг другу, то процесс поиска неисправности сосредоточивается на проблемах сети, препятствующих правильному перенаправлению пакетов.

В частности, данный раздел рассматривает команды `ping` и `traceroute`. Команда `ping` отвечает на простой вопрос: могут ли два хоста обмениваться пакетами друг с другом или нет? Команда `traceroute` использует более диагностический подход: если хосты не могут обмениваться пакетами, то она помогает сетевому инженеру определить, где кроется проблема.

Данный раздел завершается подробностями применения Telnet для перемещения между маршрутизаторами Cisco, чтобы сетевым инженерам было проще использовать команды `ping` и `traceroute`.

Команда ping

Команда ping предназначена для проверки подключения. Она посылает серию пакетов по указанному IP-адресу получателя. Данные пакеты запрашивают: "Если вы получили этот пакет, верните ответ". Каждый раз, когда отправитель посылает запрос, а другой хост посылает ответ, команда ping устанавливает, что пакет нормально прошел от хоста отправителя на хост получателя и вернулся назад.

Более формально, команда ping использует *протокол управляющих сообщений Интернета* (Internet Control Message Protocol — ICMP), а именно сообщения эхо-запроса и эхо-ответа ICMP. Протокол ICMP определяет также много других сообщений, но эти два сообщения специально предназначены для проверки подключения такими командами, как ping. Протокол ICMP не полагается на протоколы транспортного уровня, такие как TCP или UDP, и не использует протокол уровня приложений. Он призван помочь протоколу IP в управлении функциями сети IP.

На рис. 18.9 приведены сообщения ICMP с заголовками IP. В данном случае пользователь на хосте A открывает приглашение к вводу команд и вводит команду ping 172.16.2.101, проверяя подключение к хосту B. Команда посылает один эхо-запрос (этап 1) и ожидает ответ; хост B получает сообщение и отсылает эхо-ответ (этап 2). Теперь команда ping может выдать пользователю сообщение, подтверждающее работоспособность соединения.

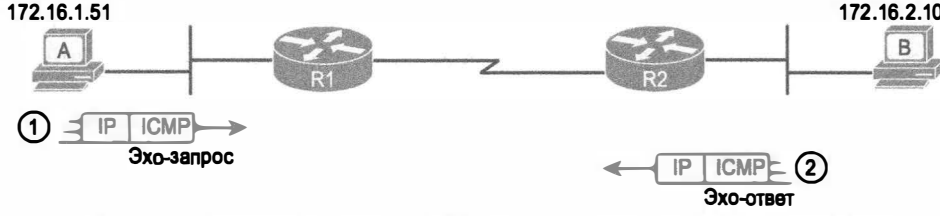


Рис. 18.9. Концепция выполнения команды ping 172.16.2.101 на хосте A

Команда ping поддерживается на многих разных устройствах и многих популярных OS. Она обладает множеством параметров: имя или IP-адрес получателя, как долго посылать эхо-запросы, как долго ожидать эхо-ответы, насколько большими сделать пакеты, и многие другие. Пример 18.9 демонстрирует вывод на хосте A той же команды ping 172.16.2.101, что и на рис. 18.9.

Пример 18.9. Пример вывода команды ping 172.16.2.101 на хосте A

```
Wendell-Odoms-iMac: wendellodom$ ping 172.16.2.101
PING 172.16.2.101 (172.16.2.101): 56 data bytes
64 bytes from 172.16.2.101: icmp_seq=0 ttl=64 time=1.112 ms
64 bytes from 172.16.2.101: icmp_seq=1 ttl=64 time=0.673 ms
64 bytes from 172.16.2.101: icmp_seq=2 ttl=64 time=0.631 ms
64 bytes from 172.16.2.101: icmp_seq=3 ttl=64 time=0.674 ms
64 bytes from 172.16.2.101: icmp_seq=4 ttl=64 time=0.642 ms
64 bytes from 172.16.2.101: icmp_seq=5 ttl=64 time=0.656 ms
^C
--- 172.16.2.101 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.631/0.731/1.112/0.171 ms
```

Проверка маршрутов IP на маршрутизаторе при помощи команды ping

Существует много инструментальных средств, позволяющих персоналу службы поддержки просмотреть даже GUI рабочего стола дистанционного пользовательского устройства. *Представители службы поддержки клиентов* (Customer Support Representatives — CSR) могут, находясь за своими компьютерами в одном городе, получить телефонный звонок от клиента из другого города и, подключившись к рабочему столу пользователя, путем нескольких щелчков установить, что проблема не сетевая. Но если проблема все-таки в сети, то дистанционное соединение с устройством пользователя может потерпеть неудачу и сотруднику CSR придется расследовать проблему подключения IP без доступа к пользовательскому устройству.

Иногда, диктуя пользователю соответствующие команды и указывая, где щелкать, можно устранить проблему. Однако пользователь может быть недоступен. В таких случаях сотрудник CSR мог бы подключиться к соседнему маршрутизатору (при помощи Telnet или SSH) и использовать команду `ping` на маршрутизаторе для проверки подключения хоста.

Предположим, например, что пользователь хоста A на рис. 18.9 обратился в службу поддержки с проблемой передачи пакетов на хост B. Попытка дистанционного подключения к рабочему столу хоста A успеха не имела. Поэтому сотрудник CSR установил сеанс Telnet с ближайшим к хосту A маршрутизатором R1 и оттуда послал эхо-запрос на хост B, как показано в примере 18.10.

Пример 18.10. Маршрутизатор R2 отправляет эхо-запрос на хост B (две команды)

```
R1# ping 172.16.2.101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.101, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = S/4 ms
R1# ping 172.16.2.101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.101, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = S/4 ms
```

Рассмотрим сначала первую команду. Команда Cisco IOS `ping` посылает пять эхо-запросов с интервалом в 2 секунды. Если команда не получает эхо-ответ в течение двух секунд, она полагает, что сообщение не прошло, и посылает следующий. Таким образом, если команда `ping` не получает ответов вообще, она выполняется примерно 10 секунд. Команда выводит точку для каждого эхо-запроса без ответа, в то время как успех (возвращение эхо-ответа в течение двух секунд) обозначается восклицательным знаком. В данном случае первый запрос не сработал, а остальные успешны.

Отметим, что команды `ping` в примере демонстрируют весьма распространенный случай: первая команда `ping` приводит несколько отказов передачи сообщения, а затем часть успешных сообщений. Обычно это происходит потому, что некое устройство на маршруте не имеет соответствующей записи в таблице ARP, и прежде чем оно будет готово перенаправить пакет, процесс ARP должен закончить работу. (Для данной проверки автор сначала использовал команду `clear ip arp 172.16.2.101` на маршрутизаторе R2 и очистил его кеш ARP, чтобы проверка показала по крайней мере один отказ.)

Контроль IP-адреса отправителя при помощи расширенной команды `ping`

Команда `ping` на маршрутизаторе R1 в примере 18.10 проверила много элементов соединения между хостами A и B, но не все. К счастью, команда IOS `ping` позволяет преодолевать эту проблему за счет использования такой функции, как расширенная команда `ping`.

Чтобы увидеть проблему (или скорее упущенную возможность), вернемся к команде `ping 172.16.2.101` на маршрутизаторе R1, показанной в примере 18.10. Команда действительно подтвердила, что многие элементы сети сработали правильно. Она непосредственно проверила физические функции и функции канала связи между маршрутизаторами R1 и R2, а также между маршрутизатором R2 и хостом B. Она проверила маршруты сетевого уровня до адреса 172.16.2.101 (прямой маршрут). Но не столь очевидная проблема в том, что она не проверяет обратный маршрут к хосту A.

Команда `ping` в примере 18.9 действительно не проверяет обратный маршрут к хосту A из-за того, что маршрутизатор выбирает IP-адрес отправителя для пакетов команды `ping`. На рис. 18.10 приведено стандартное поведение команды `ping` на маршрутизаторе. Он должен извлечь IP-адрес отправителя и использовать его для эхо-запроса, но маршрутизаторы извлекают *IP-адрес исходящего интерфейса*. Эхо-запрос от маршрутизатора R1 к хосту B передается с IP-адресом отправителя 172.16.4.1 (IP-адрес интерфейса S0/0/0 маршрутизатора R1) и, конечно, с IP-адресом получателя 172.16.2.101.

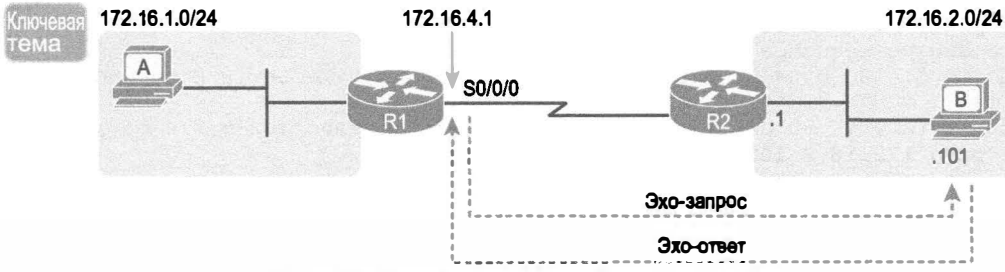


Рис. 18.10. Стандартная команда `ping 172.6.2.101` использует IP-адрес исходящего интерфейса

Упущенную возможность демонстрирует пакет эхо-ответа ICMP. Он передается на IP-адрес в подсети 172.16.4.0/24, а не на IP-адрес 172.16.1.0/24 в подсети хоста A. Таким образом, эхо-ответ ICMP не проверяет маршрут назад в подсеть хоста A. Для лучшей проверки следовало бы использовать в качестве отправителя эхо-запроса IP-адрес маршрутизатора R1 сети LAN, чтобы ответное сообщение передавалось назад в подсеть хоста A, проверяя маршруты к этой подсети так, как показано на рис. 18.11.

Расширенная команда маршрутизатора `ping` позволяет использовать несколько дополнительных параметров по сравнению со стандартной командой `ping`. Как уже можно предположить, эта команда позволяет задать как IP-адрес отправителя любой из собственных IP-адресов локального маршрутизатора. Команда может непосредственно задать IP-адрес или IP-адрес интерфейса маршрутизатора.

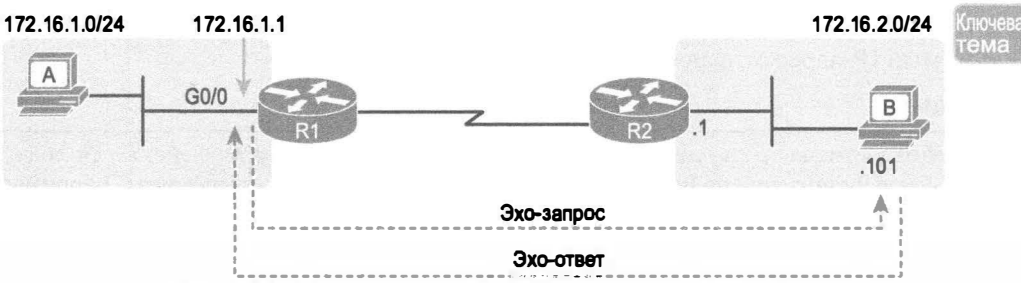


Рис. 18.11. Расширенная команда `ping` *использует LAN как IP-адрес исходящего интерфейса*

Хотя расширенная команда `ping` позволяет пользователю ввести все параметры в потенциально достаточно длинной командной строке, она позволяет также просто ввести команду `ping`, нажать клавишу <Enter>, а операционная система IOS попросит пользователя ответить на вопросы, чтобы завершить команду, как показано в примере 18.11. Пример демонстрирует команду `ping` на маршрутизаторе R1 в соответствии с логикой на рис. 18.11.

Пример 18.11. Расширенная команда `ping`, использующая LAN как IP-адрес исходящего интерфейса

```
R1# ping
Protocol [ip]:
Target IP address: 172.16.2.101
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.101, timeout is 2 seconds:
Packet sent with a source address of 172.16.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = S/4 ms
```

Читая сверху вниз, первые две выделенные строки демонстрируют начало расширенных возможностей команды `ping`. Если бы пользователь ответил **`n`** или **`no`** на вопрос `Extended commands`, то операционная система IOS не задавала бы остальные вопросы, присущие расширенной версии команды. При ответе **`yes`** она задает дополнительные вопросы, включая `Source address or interface` (Адрес отправителя или интерфейс), на который в данном случае последовал ответ **`172.16.1.1`** как IP-адрес отправителя.

Последняя выделенная строка подтверждает IP-адрес отправителя, используемый для данной проверки. Обратите внимание, что операционная система IOS вы-

водит эту строку только тогда, когда расширенная команда ping определяет используемый IP-адрес отправителя.

ВНИМАНИЕ!

Расширенная команда ping действительно позволяет проверить интерфейс, но ей должно быть предоставлено полностью квалифицированное имя интерфейса, например gigabitethernet0/0.

Расширенная версия команды ping, использующей параметр исходящего интерфейса, обеспечивает более реалистичную проверку соединения. При сравнении в этом разделе стандартных и расширенных примеров команды ping обе выполняют ту же задачу по проверке подключения и каналов связи между маршрутизатором R1 и хостом B. Однако расширенная команда ping проверяет вероятный маршрут назад к пользовательскому хосту (хосту A), в то время как стандартная команда ping не делает этого.

Команда traceroute

Вообразите какого-либо сетевого инженера или сотрудника CSR, приступившего к исследованию некоторой проблемы. Он вводит команду ping для хоста пользователя, для соседнего маршрутизатора и после нескольких попыток убеждается, что хост действительно может посылать и получать пакеты IP. Проблема еще не устранена, но уже понятно, что она не связана с передачей пакетов между устройством пользователя и остальной частью сети.

Теперь представим следующую проблему, когда команда ping для устройства потерпела неудачу. Это свидетельствует о том, что в сети IP действительно существует некая проблема. Но где она? На что инженер должен обратить пристальное внимание? Хотя команда ping весьма и весьма полезна, команда traceroute обладает лучшими возможностями по уточнению причины проблемы. Команда traceroute позволяет уточнить источник проблемы, демонстрируя, как далеко проходит пакет по сети IP, прежде чем произойдет отказ.

Команда traceroute, будучи выполненной полностью, идентифицирует все маршрутизаторы по пути от хоста отправителя до хоста получателя. В частности, она выводит IP-адрес следующего транзитного пункта каждого маршрутизатора, указанного в каждом индивидуальном маршруте. Например, команда traceroute 172.16.2.101 на хосте A идентифицировала бы IP-адрес маршрутизатора R1, затем маршрутизатора R2 и наконец хоста B, как показано на рис. 18.12. В примере 18.12 показан вывод этой команды, введенной на хосте A.

Пример 18.12. IP-адреса, идентифицированные при успешном выполнении команды traceroute 172.16.2.101 на хосте A

```
Wendell-Odoms-iMac: wendellodom$ traceroute 172.16.2.101
traceroute to 172.16.2.101, 64 hops max, 52 byte packets
  1 172.16.1.1 (172.16.1.1) 0.870 ms 0.520 ms 0.496 ms
  2 172.16.4.2 (172.16.4.2) 8.263 ms 7.518 ms 9.319 ms
  3 172.16.2.101 (172.16.2.101) 16.770 ms 9.819 ms 9.830 ms
```

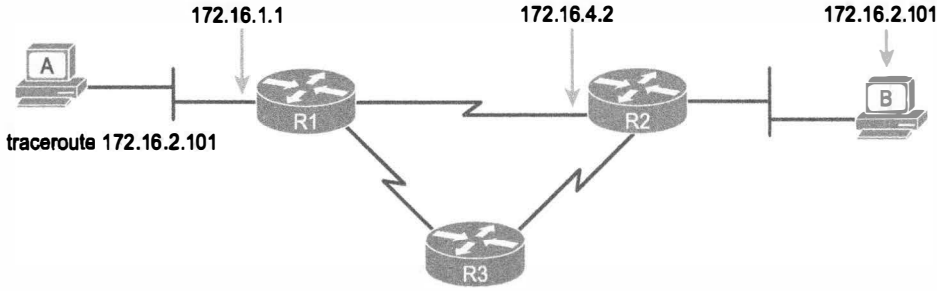


Рис. 18.12. IP-адреса, идентифицированные при успешном выполнении команды `traceroute 172.16.2.101` *на хосте A*

Как работает команда `traceroute`

Команда `traceroute` собирает эту информацию по каждому маршрутизатору на основании сообщения ICMP, первоначально предназначенного для совершенно других целей: сообщения *превышения времени существования* (Time-to-Live Exceeded — TTL) протокола ICMP.

Прежде чем обсуждать команду `traceroute`, имеет смысл ознакомиться с сообщениями TTL и TTL Exceeded. В сети IP маршрутизаторы могут создать цикл. Циклический маршрут — это то, чего следует избегать, поскольку, передавая пакеты по кругу, маршрутизаторы никогда не доставят их получателю. Предположим, например, что маршрутизатор R1 посылает пакет маршрутизатору R2, который посылает его маршрутизатору R3, который снова посылает его маршрутизатору R1, и так снова и снова по кругу между этими тремя маршрутизаторами.

Маршрутизаторы Ipv4 сталкиваются с одним очень неприятным побочным эффектом петлевых маршрутов, из-за которого пакеты бесконечно блуждали бы по сети. Для ликвидации зацикленных пакетов IP заголовок Ipv4 содержит поле TTL (Time-to-Live *время существования*). Это значение устанавливает хост отправителя. Впоследствии каждый маршрутизатор, передающий пакет, увеличивает значение TTL на 1. Когда значение поля TTL достигает 0, маршрутизатор понимает, что пакет зациклен, и отбрасывает его. Маршрутизатор также уведомляет хост, пославший пакет, об отказе его перенаправления передачей сообщения TTL Exceeded (превышение времени существования) протокола ICMP.

Теперь вернемся к команде `traceroute`. Она посылает сообщения так, чтобы заставить маршрутизаторы возвращать сообщения TTL Exceeded даже без наличия в сети циклического маршрута. В результате команда `traceroute` способна идентифицировать маршрутизатор на основании IP-адреса отправителя пакета, содержащего сообщение TTL Exceeded.

Для этого команда `traceroute` начинает с передачи нескольких пакетов (обычно трех), поле TTL заголовка которых содержит значение 1. Когда такой пакет достигает следующего маршрутизатора (в данном примере стандартного маршрутизатора R1 хоста A), декремент поля TTL дает 0 и маршрутизатор отбрасывает пакет. В результате маршрутизатор посылает хосту A сообщение TTL Exceeded, позволяющее команде `traceroute` идентифицировать IP-адрес маршрутизатора. Один из таких пакетов и сообщение TTL Exceeded представлены на рис. 18.13.

Ключевая тема

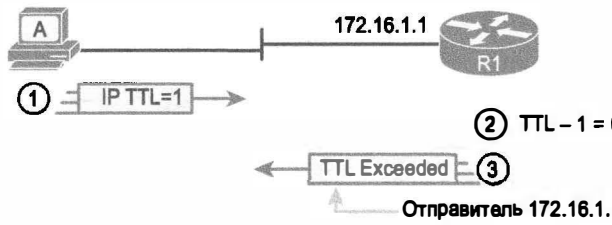


Рис. 18.13. Как команда `traceroute` идентифицирует первый маршрутизатор на маршруте

Команда `traceroute` посылает несколько пакетов TTL=1, ожидая ответных сообщений TTL Exceeded от того же маршрутизатора, идентифицируемого на основании IP-адреса отправителя в сообщениях TTL Exceeded. Учитывая, что все сообщения исходят от того же маршрутизатора, команда `traceroute` отображает его IP-адрес в отдельной строке вывода. У маршрутизаторов есть выбор используемых IP-адресов, но, как уже можно предположить, они используют IP-адрес исходящего интерфейса. В данном случае исходящий интерфейс маршрутизатора R1 для сообщения имеет адрес 172.16.1.1.

Для поиска всех маршрутизаторов по пути и окончательного подтверждения прохождения пакетов к хосту получателя команда `traceroute` посылает пакеты TTL=2, затем 3, 4 и т.д., пока не ответит хост получателя. На рис. 18.14 представлен первый пакет TTL=2, демонстрирующий, как маршрутизатор R1 фактически перенаправляют пакет, а декремент значения поля TTL на маршрутизаторе R2 до 0 заставляет его отослать сообщение TTL Exceeded назад на хост A.

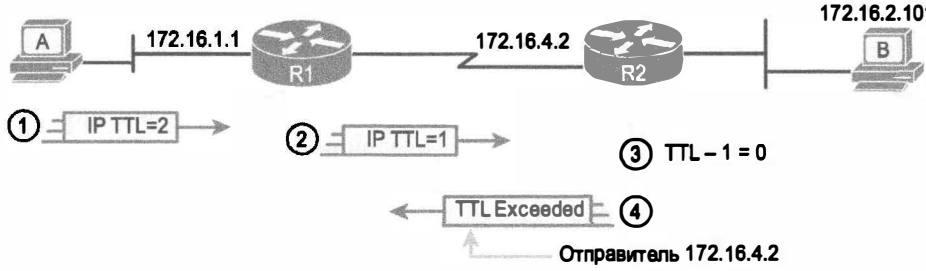


Рис. 18.14. Сообщения TTL=2, посланные командой `traceroute`

На рисунке показаны следующие четыре этапа.

Этап 1 Команда `traceroute` посылает пакет со значением 2 в поле TTL

Этап 2 Маршрутизатор R1 обрабатывает пакет и уменьшает значение поля TTL до 1. Маршрутизатор R1 перенаправляют пакет

Этап 3 Маршрутизатор R2 обрабатывает пакет и уменьшает значение поля TTL до 0. Маршрутизатор R2 отбрасывает пакет

Этап 4 Маршрутизатор R2 уведомляет хост отправителя пакета об отбрасывании пакета сообщением TTL Exceeded протокола ICMP. IP-адресом отправителя этого сообщения является адрес исходящего интерфейса маршрутизатора R2, в данном случае 172.16.4.2

Если команда traceroute завершает работу и ее вывод содержит в последней строке IP-адрес хоста получателя, все прекрасно! В этом случае команда подтвердила наличие соединения между хостом отправителя и получателя в обоих направлениях. Но если есть проблема маршрутизации IP, команда может продолжить выполнение, пока пользователь не отменит ее или пока она не потерпит неудачу на одном из этапов.

Если команда traceroute не может завершить работу или терпит неудачу, несколько последних строк вывода позволят существенно сэкономить время поиска и устранения неисправности. В этом случае проблема, вероятней всего, возникла на последнем указанном в выводе маршрутизаторе или на следующем, который должен был быть указан.

Например, если приведенная ранее команда traceroute 172.16.2.101 укажет в выводе маршрутизатор R1 (172.16.1.1), но не другие маршрутизаторы, то поиск и устранение неисправностей следует осуществлять на маршрутизаторах R1 и R2. В большой сети сужение области поиска проблемы до нескольких маршрутизаторов может быть отличным первым шагом в исследовании ее причины.

Команда traceroute и подобные ей

Команды ping и traceroute поддерживает большинство операционных систем, включая Cisco IOS. Однако некоторые OS используют немного иной синтаксис команды traceroute. Например, в большинстве OS Windows используются команды tracert и pathping, а не traceroute. Операционные системы Linux и MAC OS X поддерживают команду traceroute.

Операционная система Cisco IOS поддерживает команду traceroute с сокращением trace. Как и команда ping, команда traceroute может быть введена со всеми параметрами в одной строке, или можно позволить IOS опросить пользователя. Подобно расширенной команде ping, расширенная команда traceroute позволяет сетевому инженеру задать IP-адрес отправителя в тех же случаях, что и команда ping.

Пример 18.13 демонстрирует вывод двух команд traceroute на маршрутизаторе R1. Первая, стандартная команда traceroute использует стандартный исходящий интерфейс, выбранный на маршрутизаторе R1. Вторая использует метод опроса операционной системой IOS и демонстрирует выбор пользователем иного IP-адреса отправителя (172.16.1.1).

Пример 18.13. Стандартная и расширенная команды traceroute на маршрутизаторе R1

```
R1# traceroute 172.16.2.101
Type escape sequence to abort.
Tracing the route to 172.16.2.101
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.4.2 0 msec 0 msec 0 msec
  2 172.16.2.101 0 msec 0 msec *

R1# traceroute
Protocol [ip]:
Target IP address: 172.16.2.101
```

```
Source address: 172.16.1.1
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 172.16.2.101
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.4.2 0 msec 0 msec 0 msec
  2 172.16.2.101 0 msec 0 msec *
```

ВНИМАНИЕ!

Команда `traceroute` ОС хоста обычно создает эхо-запросы ICMP. Команда `traceroute` операционной системы Cisco IOS, напротив, создает пакеты IP с заголовком UDP. В настоящее время эта информация может показаться незначительной, но при обсуждении списков управления доступом (ACL) будет продемонстрировано, что они фактически способны фильтровать трафик сообщений `traceroute` хоста, но не маршрутизатора, и наоборот.

Telnet и приостановка

Большинство сетевых инженеров исследуют сетевые проблемы, сидя за своим компьютером. Чтобы получить доступ к маршрутизатору или коммутатору, инженеру достаточно установить сеанс Telnet или SSH со своего компьютера и подключиться к необходимому маршрутизатору или коммутатору. Как правило, открывают несколько окон Telnet или SSH для подключения к нескольким устройствам.

В качестве альтернативы инженер мог бы подключиться к маршрутизатору или коммутатору, используя клиент Telnet или SSH на своем компьютере, а затем использовать пользовательские команды Cisco IOS `telnet` или `ssh` для подключения к другим маршрутизаторам и коммутаторам. Эти команды действуют как клиент Telnet или SSH соответственно, поэтому при поиске и устранении неисправностей можно легко подключиться к другим устройствам. По завершении пользователю достаточно использовать команду `exit`, чтобы отключить сеанс Telnet или SSH.

Одним из важнейших преимуществ использования команд Cisco IOS `telnet` и `ssh` является средство *приостановки* (suspend). Приостановка позволяет соединению Telnet или SSH оставаться активными при создании другого соединения Telnet или SSH, чтобы можно было создать несколько параллельных соединений, а затем легко переключаться между ними. На рис. 18.15 представлена типичная объединенная сеть, на примере которой демонстрируется вся мощь средства приостановки.

Администратор маршрутизатора использует компьютер по имени Bench для установки сеанса Telnet с маршрутизатором Cincy. Подключившись к интерфейсу CLI маршрутизатора Cincy, но устанавливает сеанс Telnet с маршрутизатором Milwaukee. Находясь на маршрутизаторе Милуоки, администратор приостанавливает сеанс Telnet, нажав комбинацию клавиш <Ctrl-Shift-6>, а затем клавишу <x>. (Обратите внимание, что комбинация <Ctrl-Shift-6> передает символ прерывания, но некоторые национальные раскладки клавиатуры могут иметь для этого иную комбинацию.) Затем администратор устанавливает сеанс Telnet из CLI маршрутизатора Cincy с мар-

шрутизатором New York и снова приостанавливает соединение. В конце примера он одновременно соединяется со всеми тремя маршрутизаторами в одном окне и получает возможность переключаться между ними лишь несколькими нажатиями клавиш. Пример 18.14 демонстрирует вывод процесса с комментариями.

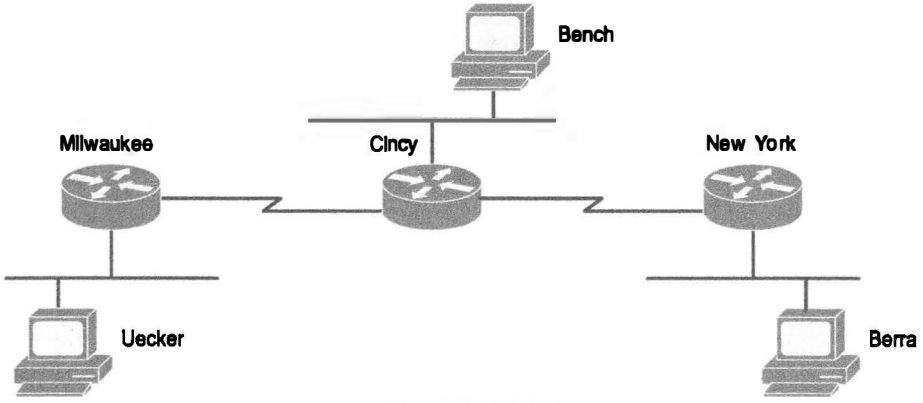


Рис. 18.15. Приостановка Telnet

Пример 18.14. Приостановка Telnet

```
Cincy# telnet milwaukee       (Ввод команды telnet для Milwaukee)
Trying Milwaukee (10.1.4.252)... Open

User Access Verification

Password:          (Ввод пароля позволит вводить команды на Milwaukee)
Milwaukee>
Milwaukee>
Milwaukee>

                          (Нажатие комбинации <Ctrl-Shift-6>, а затем <x>)
Cincy# telnet NewYork     (Возврат к Cincy, поскольку Telnet приостановлен)


Trying NewYork (10.1.6.253)... Open


     (Теперь вход на New York, на основании команды telnet NewYork)
User Access Verification
Password:
NewYork>                  (Теперь можно ввести команды на NewYork)
NewYork>
NewYork>
NewYork>

                           (Нажатие комбинации <Ctrl-Shift-6>, а затем <x>)
                           (В результате возврат на маршрутизатор Cincy)
Cincy# show sessions (Эта команда выводит приостановленные сеансы Telnet)
Conn Host       Address     Byte  Idle Conn Name
   1 Milwaukee  10.1.4.252    0     0   Milwaukee
*  2 NewYork    10.1.6.253    0     0   NewYork
```

```
Cincy# where (where does the same thing as show sessions)
Conn Host       Address     Byte  Idle Conn Name
   1 Milwaukee  10.1.4.252    0    0    Milwaukee
*  2 NewYork    10.1.6.253    0    0    NewYork

Cincy# resume 1 (Возобновить соединение 1 (см. Show session) с Milwaukee)
[Resuming connection 1 to milwaukee ... ]

Milwaukee>          (Теперь можно ввести команды на Milwaukee)
Milwaukee>
Milwaukee>
! (Нажатие комбинации <Ctrl-Shift-6>, а затем <x>, поскольку нужно
! вернуться на Cincy)

Cincy#  (ОГО! Просто нажал <Enter> и возобновил последний сеанс Telnet)
[Resuming connection 1 to milwaukee ... ]

Milwaukee>
Milwaukee>
Milwaukee>

                    (Нажатие комбинации <Ctrl-Shift-6>, а затем <x>)
                    (В результате возврат на маршрутизатор Cincy)
Cincy# disconnect 1 (Telnet Milwaukee больше не нужен, остановить!)
Closing connection to milwaukee [confirm] (Нажать <Enter> и подтветдить)
Cincy#
[Resuming connection 2 to NewYork ... ]
          (Нажатие <Enter> возобновит последний активный сеанс Telnet)

NewYork>
NewYork>
NewYork>

                    (Нажатие комбинации <Ctrl-Shift-6>, а затем <x>)
                    (В результате возврат на маршрутизатор Cincy)
Cincy# disconnect 2 (Telnet NewYork больше не нужен, остановить!)
Closing connection to NewYork [confirm] (Нажать <Enter> и подтветдить)
Cincy#
```

Примечания в примере объясняют большинство деталей. Пример 18.14 начинается с приглашения к вводу команд маршрутизатора Cincy, которое было бы в окне Telnet на хосте Bench. После установления соединения Telnet с маршрутизатором Milwaukee оно было приостановлено, поскольку администратор нажал комбинацию клавиш <Ctrl-Shift-6>, а затем нажал клавишу <x>. Далее, после установления соединения Telnet с маршрутизатором New York, оно также было приостановлено той же комбинацией клавиш.

Приостановить и возобновить эти два соединения легко. Команда `resume` возобновит любое приостановленное соединение. Для возобновления конкретного сеанса команда `resume` может использовать идентификатор соединения, представляемый командой `show sessions`. (Команда `where` обеспечивает тот же вывод.) Если команда `resume` используется без идентификатора соединения, она возобновляет последний приостановленный сеанс. Кроме того, вместо команды `resume`

можно использовать только номер сеанса. Например, ввод команды `2` даст тот же результат, что и ввод команды `resume 2`.

Вот интересный, но потенциально опасный нюанс: если сеанс Telnet приостановлен и пользователь просто нажал клавишу <Enter>, программное обеспечение Cisco IOS возобновит последнее приостановленное соединение Telnet. Звучит безобидно, пока не поймешь, что имеешь привычку нажимать клавишу <Enter> для очистки экрана. При приостановленном сеансе Telnet несколько нажатий клавиши <Enter> не столько очистит экран, сколько возобновит соединение с другим маршрутизатором. Это особенно опасно при изменении конфигурации или использовании потенциально деструктивных команд. Поэтому будьте внимательны и следите за фактически используемым маршрутизатором, когда имеете приостановленные соединения Telnet.

Если необходимо узнать, какой сеанс был приостановлен последним, ищите в выводе команды `show sessions` сеанс со звездочкой (`*`) слева от записи. Звездочкой отмечен последний приостановленный сеанс.

Кроме команд в примере 18.14, позволяющих приостановить и возобновить соединение Telnet или SSH, есть две другие команды, способные предоставить полезную информацию о сеансах для пользователей, зарегистрированных на маршрутизаторе. Команда `show users` выводит всех пользователей, зарегистрированных на маршрутизаторе, на котором использована команда. Она перечисляет все сеансы, включая как пользователей с консоли, так и подключившихся по Telnet или SSH. Команда `show ssh` выводит ту же информацию, но для пользователей, подключившихся по SSH. Обратите внимание, что эти команды отличаются от команды `show sessions`, выводящей приостановленные сеансы Telnet и SSH с локального маршрутизатора на другие устройства.

Обзор

Резюме

- Протокол динамического конфигурирования хоста (DHCP) является одним из наиболее популярных протоколов в сети TCP/IP. Подавляющее большинство хостов в сети TCP/IP — это пользовательские устройства, а подавляющее большинство пользовательских устройств изучает свои параметры Ipv4, используя протокол DHCP.
- В процессе предоставления IP-адреса протоколом DHCP между клиентом и сервером передаются следующие четыре сообщения. (Чтобы проще запомнить эти сообщения, их первые символы складываются в слово DORA.)
  - *Discover* (обнаружить). Передается клиентом DHCP при поиске подходящего сервера DHCP.
  - *Offer* (предложение). Передается сервером DHCP как предложение клиенту некого IP-адреса (и сообщение других параметров).
  - *Request* (запрос). Передается клиентом DHCP как запрос серверу на резервирование Ipv4-адреса, указанного им в сообщении Offer.
  - *Acknowledgment* (подтверждение). Передается сервером DHCP при назначении адреса и информировании о маске, а также IP-адресах стандартного маршрутизатора и сервера DNS.
- У клиентов DHCP есть специфическая проблема: у них еще нет IP-адреса, но они должны посылать пакеты IP. Для решения этой проблемы сообщения DHCP используют два специальных Ipv4-адреса, позволяющих хосту без IP-адреса посылать и получать сообщения по локальной подсети.
  - `0.0.0.0`. Специально зарезервированный Ipv4-адрес для использования хостами, у которых еще нет IP-адреса.
  - `255.255.255.255`. Адрес, зарезервированный как широковещательный адрес локальной подсети. Посланные на этот адрес пакеты распространяются по локальному каналу связи, но в другие подсети маршрутизаторы их не перенаправляют.
- Большинство корпоративных сетей используют несколько серверов DHCP на централизованной площадке, обеспечивающих услуги протокола DHCP для всех дистанционных подсетей. Маршрутизаторы так или иначе должны перенаправлять эти сообщения DHCP между клиентами и сервером DHCP. Для этого маршрутизаторы, соединенные с дистанционными подсетями LAN, нуждаются в подкоманде интерфейса `ip helper-address` *`IP-адрес_сервера`*.
- Ниже приведены типы параметров, которые должны быть известны серверу DHCP для поддержки клиентов DHCP.
  - *Идентификатор подсети и маска.* Сервер DHCP может использовать эту информацию для вычисления всех адресов в подсети. Обычно сервер способен предоставить любой допустимый адрес в подсети, кроме зарезервированных и исключенных. (Серверу DHCP известно, что нельзя предоставлять идентификатор и широковещательный адрес подсети.)

- *Зарезервированные (исключенные) адреса*. Серверу должны быть известны не предоставляемые адреса подсети. Этот список позволяет исключить IP-адреса, присваиваемые только статически. Например, статически присваивается большинство IP-адресов маршрутизаторов и коммутаторов, серверов и многих других устройств, кроме пользовательских. Как правило, инженеры используют то же соглашение для всех подсетей, резервируя либо самые младшие IP-адреса во всех подсетях, либо самые старшие.
- *IP-адрес стандартного маршрутизатора*. IP-адрес маршрутизатора данной подсети.
- *IP-адрес сервера DNS*. Список IP-адресов серверов DNS.

- Вот этапы настройки сервера DHCP на базе Cisco IOS.

  Этап 1 Исключить адреса из набора предоставляемых сервером DHCP: `ip dhcp excluded-address` *первый последний*

  Этап 2 Создать пул DHCP и перейти в режим конфигурации пула: `ip dhcp pool` *имя*

  A. Определить подсеть, поддерживаемую сервером DHCP: `network` *идентификатор_подсети маска* или `network` *идентификатор_подсети длина_префикса*

  B. Определить IP-адрес (адреса) стандартного маршрутизатора в данной подсети: `default-router` *адрес1 адрес2...*

  C. Определить список IP-адресов сервера DNS: `dns-server` *адрес1 адрес2...*

  D. Определить продолжительность предоставления в днях, часах и минутах: `lease` *дней часов минут*

  E. Определить имя домена DNS: `domain-name` *имя*

- У сервера DHCP IOS есть несколько других команд `show`. Три из них приведены ниже.
  - `show ip dhcp binding`. Выводит информацию обо всех IP-адресах, предоставленных клиентам в настоящее время.
  - `show ip dhcp pool [`*имяпула*`]`. Выводит заданный диапазон IP-адресов, а также количество предоставленных в настоящее время адресов и максимальное количество резервируемых адресов по каждому пулу.
  - `show ip dhcp server statistics`. Выводит статистику сервера DHCP.
- Чтобы стандартный маршрутизатор хоста LAN работал правильно, для него должно быть истинно следующее.
  - Канал связи хоста с локальной сетью и канал связи стандартного маршрутизатора с локальной сетью должны находиться в той же сети VLAN.
  - IP-адреса хоста и стандартного маршрутизатора должны принадлежать той же подсети.
  - Настроенный на хосте адрес стандартного маршрутизатора должен совпадать с IP-адресом, настроенным на маршрутизаторе.
  - Коммутаторы LAN не должны отбрасывать фреймы из-за настроек защиты порта.
- Команда `ping` предназначена для проверки подключения. Она посылает серию пакетов по указанному IP-адресу получателя. Данные пакеты запрашивают: "Если вы получили этот пакет, верните ответ". Каждый раз, когда от-

правитель посылает запрос, а другой хост посылает ответ, команда `ping` устанавливает, что пакет нормально прошел от хоста отправителя на хост получателя и вернулся назад.

- Команда `traceroute`, будучи выполненной полностью, идентифицирует все маршрутизаторы по пути от хоста отправителя до хоста получателя. В частности, она выводит IP-адрес следующего транзитного пункта каждого маршрутизатора, указанного в каждом индивидуальном маршруте.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Компьютер подключен к сети LAN и использует протокол DHCP для резервирования IP-адреса в первый раз. Какое из четырех сообщений DHCP, передаваемых обычно между компьютером и сервером DHCP, посылает клиент? (Выберите два ответа.)

   А) Acknowledgment.

   Б) Discover.

   Г) Offer.

   Д) Request.

2. Предприятие располагает серверы DHCP и DNS в сети VLAN 10 (подсети 10) на маршрутизаторе Atlanta, использующем IP-адрес 10.1.10.1 для сервера DHCP и IP-адрес 10.1.10.2 для сервера DNS. Дистанционный маршрутизатор Boston находится в Бостоне, а устройства его LAN используют серверы DHCP и DNS в Атланте. Какая из следующих команд должна быть настроена на маршрутизаторах в этом предприятии, чтобы обеспечить поддержку служб DHCP и DNS?

   А) Команда `ip helper-address 10.1.10.1` на маршрутизаторе Atlanta.

   Б) Команда `ip helper-address 10.1.10.2` на маршрутизаторе Boston.

   В) Команда `ip name-server 10.1.10.2` на маршрутизаторе Atlanta.

   Г) Команда `ip dhcp-server 10.1.10.1` на маршрутизаторе Boston.

   Д) Ни один из ответов не правильный.

3. Фред решает перейти со старой платформы сервера DHCP на использование маршрутизатора Cisco в здании штаб-квартиры. Этот сервер DHCP, созданный на базе операционной системы IOS маршрутизатора Cisco, поддерживает 200 дистанционных подсетей. Какие из следующих параметров следует установить вне пула адресов каждой подсети?

   А) IP-адрес клиента.

   Б) Адреса в данной подсети, исключенные из списка предоставляемых сервером.

   В) Адрес стандартного маршрутизатора.

   Г) Адрес сервера DNS.

   Д) Длина резервируемого адреса.

4. Компьютер PC1 использует статическую (заданную вручную) конфигурацию Ipv4, связанную с его сетевой платой Ethernet. Какой из следующих параметров Ipv4 компьютер укажет в своей конфигурации Ipv4? (Выберите два ответа.)

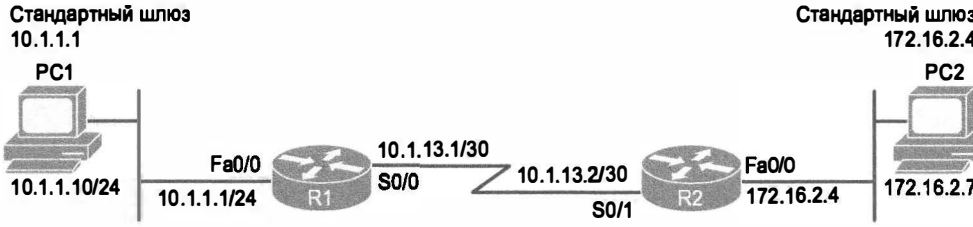
А) Адрес сервера DHCP.

Б) Адрес сервера DNS.

В) Адрес сервера трассировки.

Г) Собственный IP-адрес компьютера.

Для ответа на вопросы 5 и 6 используйте следующий рисунок.



5. Новый сетевой инженер пытается выяснить причину проблем пользователя компьютера PC1. Результаты каких из следующих действий вероятней всего выявили бы причину проблем на уровнях 1 или 2 в сети LAN Ethernet, показанной слева на рисунке?

   А) Команда `ping 10.1.1.1` на компьютере PC1 не имела успеха.

   Б) Команда `ping 10.1.13.2` на компьютере PC1 успешна, а команда `ping 172.16.2.4` — нет.

   В) Команда `ping 10.1.1.1` на компьютере PC1 успешна, а команда `ping 10.1.13.1` — нет.

   Г) Команда `ping 10.1.1.10` на компьютере PC1 успешна.

6. Пользователь компьютера PC2 ввел команду `tracert 10.1.1.10`. Какой из следующих IP-адресов мог бы быть представлен в выводе команды? (Выберите три ответа.)

   А) 10.1.1.10.

   Б) 10.1.1.1.

   В) 10.1.13.1.

   Г) 10.1.13.2.

   Д) 172.16.2.4.

7. Рассмотрите следующий вывод команды. Что было бы, введи пользователь команду `resume`?

```
R1# show sessions
Conn Host  Address  Byte Idle Conn Name
 1 Fred  10.1.1.1 0  0  Fred
* 2 Barney 10.1.2.1 0  0  Barney
```

   А) Команда будет отклонена, а CLI снова отобразит приглашение к вводу команд маршрутизатора R1.

   Б) Пользователь CLI возобновит приостановленное соединение Telnet с маршрутизатором по адресу 10.1.1.1.

   В) Пользователь CLI возобновит приостановленное соединение Telnet с маршрутизатором по адресу 10.1.2.1.

   Г) Представленной информации недостаточно для точного предсказания результата.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 18.1.

Таблица 18.1. Ключевые темы главы 18

Элемент	Описание	Страница
Список	Четыре сообщения DHCP, используемых при предоставлении нового IP-адреса	539
Список	Специальные Ipv4-адреса 0.0.0.0 и 255.255.255.255	539
Список	Четыре логических этапа, обусловленных командой ip helper-address	541
Рис. 18.2	Что команда ip helper-address изменяет в сообщении DHCP Discover (эффект вспомогательного IP-адреса)	541
Список	Этапы настройки сервера DHCP на базе Cisco IOS	544
Список	Последовательность проверки соответствия параметров хоста Ipv4 параметрам стандартного маршрутизатора Ipv4	551
Рис. 18.10	Стандартная команда ping 172.6.2.101 использует IP-адрес исходящего интерфейса	556
Рис. 18.11	Расширенная команда ping использует LAN как IP-адрес исходящего интерфейса	557
Рис. 18.13	Как команда traceroute идентифицирует первый маршрутизатор на маршруте	560

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

Клиент DHCP (DHCP client), сервер DHCP (DHCP server), широковещательный адрес локальной подсети (local subnet broadcast address), ping, traceroute, ретранслятор DHCP (DHCP relay), ICMP

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 18.2 приведен список команд конфигурации, а в табл. 18.3 приведены пользовательские команды главы. Фактически команды стоит запомнить, чтобы лучше понять материал главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 18.2. Команды конфигурации главы 18

Команда	Описание
`ip dhcp exclude-address` *первый последний*	Глобальная команда, исключающая диапазон адресов из предоставляемых сервером DHCP
`ip dhcp pool` *имяпула*	Глобальная команда, создающая пул по имени *имяпула* и переводящая пользователя в режим конфигурации пула сервера DHCP
`network` *идентификатор_подсети* {*маска-ddn* | */длина_префикса*}	Подкоманда режима пула DHCP, определяющая сеть или подсеть, требующую от сервера DHCP предоставления IP-адресов
`default-router` *адрес1 адрес2…*	Подкоманда режима пула DHCP, определяющая один или несколько маршрутизаторов как стандартные маршрутизаторы
`dns-server` *адрес1 адрес2…*	Подкоманда режима пула DHCP, определяющая список серверов DNS, которые указал сервер DHCP
`lease` *дней часов минут*	Подкоманда режима пула DHCP, определяющая период резервирования DHCP
`ip helper-address` *ip-адрес*	Подкоманда интерфейса, указывающая маршрутизатору обращать внимание на широковещательные адреса локальной подсети (255.255.255.255) и менять IP-адреса отправителя и получателя, позволяя серверам DHCP находиться в дистанционной подсети
`ip name-server` *адрес1…*	Глобальная команда, указывающая маршрутизатору или коммутатору выводить список IP-адресов серверов DNS, что позволяет пользователям CLI этого маршрутизатора или коммутатора преобразовывать имена хостов, используя DNS
`ip host` *имя адрес*	Глобальная команда, позволяющая статически сопоставить имя с IP-адресом хоста, чтобы маршрутизатор или коммутатор не должен был запрашивать преобразование имени на сервере DNS
`[no] ip domain-lookup`	Глобальная команда, включающая или отключающая функцию преобразователя DNS на маршрутизаторе или коммутаторе
`ip address` *ip-адрес маска* `[secondary]`	Подкоманда интерфейса, присваивающая IP-адрес интерфейса и (необязательно) делающая адрес вторичным

Таблица 18.3. Пользовательские команды главы 18

Команда	Описание
`show arp, show ip arp`	Выводит таблицу ARP маршрутизатора Ipv4
`show ip dhcp binding`	Выводит зарезервированные в настоящий момент IP-адреса на сервере DHCP, а также идентификатор клиента и информацию о периоде резервирования

Окончание табл. 18.3

Команда	Описание
`show ip dhcp pool` *имя*	Выводит заданный диапазон адресов в пуле наряду с количеством предоставленных в настоящее время адресов и максимальным количеством резервируемых адресов по каждому пулу
`show ip dhcp server statistics`	Выводит статистику запросов сервера DHCP
`show ip dhcp conflict`	Выводит IP-адреса, обнаруженные сервером DHCP, как уже занятые при попытке предоставить их хосту
`clear ip dhcp conflict`	Удаляет все записи из списка конфликтных адресов сервера DHCP
`ping {`*имя_хоста* `|` *ip-адрес*`}`	Проверяет маршрут IP, посылая пакеты ICMP на хост получателя
`traceroute {`*имя_хоста* `|` *ip-адрес*`}`	Проверяет маршрут IP, обнаруживая IP-адреса промежуточных маршрутизаторов и выводя их получателю
`telnet {`*имя_хоста* `|` *ip-адрес*`}`	Создает соединение Telnet от локального маршрутизатора или коммутатора с указанным в команде хостом
`show sessions where`	Выводит соединения Telnet или SSH от локального маршрутизатора с другим устройством, но в настоящее время приостановленные. Зная приостановленные соединения, пользователи могут выбирать и возобновлять их
`resume` *номер_сеанса*	Возобновление приостановленного ранее соединения Telnet или SSH от некоего маршрутизатора или коммутатора с другим устройством
`disconnect` *номер_сеанса*	Отключает предварительно приостановленное соединение Telnet или SSH от некоего маршрутизатора или коммутатора с другим устройством
`show users`	Выводит список всех пользователей, зарегистрированных в настоящее время на маршрутизаторе или коммутаторе

Таблица 18.4. Команды работы с сетевыми хостами главы 18

Команда	Описание
`ipconfig, ifconfig`	Выводит список параметров IP для интерфейса (сетевой платы)
`nslookup` *имя*	Отправляет запрос на преобразование имен и выводит результат
`netstat -rn`	Выводит таблицу маршрутизации хоста; стандартный маршрутизатор обычно указывается как маршрут к 0.0.0.0
`arp -a`	Выводит таблицу ARP хоста
`ping {`*имя_хоста* `|` *ip-адрес*`}`	Проверяет маршрут IP, посылая пакеты ICMP на хост получателя
`traceroute	tracert	pathping {`*имя_хоста* `|` *ip-адрес*`}`	Проверяет маршрут IP, обнаруживая IP-адреса промежуточных маршрутизаторов и выводя их получателю

Ответы на контрольные вопросы:

1 Г и Д. **2** В и Г. **3** Б, В, Г, и Д. **4** Б. **5** А и Г. **6** Б. **7** А. **8** А и Д.

Обзор части IV

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части IV

Задача	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей команд по категориям		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав только этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения к книге.

Ответы на вопросы

Ответьте на вопросы обзора этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров только этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения к книге.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если понятны не все их подробности, уделите время повторному изучению.

Создайте диаграмму связей команд по категориям

Как и в части II, в части IV представлено множество новых команд CLI, на сей раз относящихся к маршрутизаторам. Такое количество команд трудно запомнить, поэтому имеет смысл выполнить специальные упражнения, позволяющие лучше понять детали и вспомнить их при необходимости.

Задача этого упражнения заключается в том, чтобы помочь запомнить команды. Оно не сосредоточивается на деталях и каждом отдельном параметре каждой команды или даже их значении. Цель в том, чтобы помочь организовать команды в памяти и помочь их вспомнить, когда они встретятся в реальности или на экзамене.

Подобно диаграмме связей части I, создайте диаграмму связей со следующими категориями команд: команды интерфейса, затрагивающие уровни 1 и 2, IP-адресация, статическая и стандартная маршрутизация, магистральное соединение маршрутизатора и коммутация третьего уровня, OSPF, сервер DHCP, проверка подключения, команды работы с сетями хоста и разное.

Кроме того, для лучшего усвоения можно также включить еще три категории для команд, относящихся и к маршрутизаторам, и к коммутаторам. Сюда относятся ко-

манды из главы 15, затронутые здесь лишь кратко и подробно описанные в части II: консоль и VTY, SSH и администрирования маршрутизатора (коммутатора).

В этой диаграмме связей для каждой категории вспомните все команды конфигурации и все пользовательские команды (главным образом, команды `show`). По каждой категории сгруппируйте команды конфигурации отдельно от пользовательских команд. Пример диаграммы связей приведен на рис. Ч4.1.

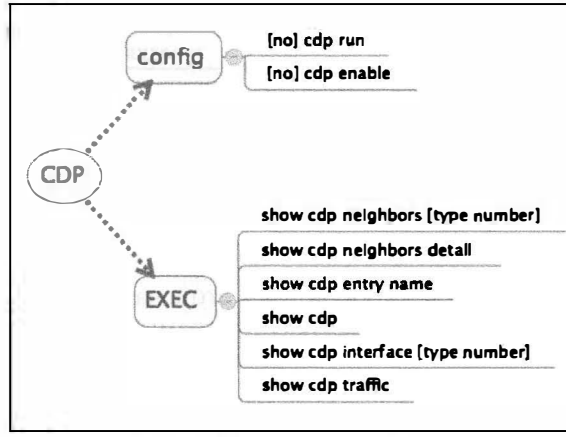


Рис. Ч4.1 Пример диаграммы связей

ВНИМАНИЕ!

Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения к книге.

И наконец, работая над этим проектом, имейте в виду следующие важные моменты.

- Основная учебная задача этого упражнения осуществляется по мере его выполнения. Чтение других диаграмм связей или только таблиц команд не настолько способствует запоминанию необходимого материала.
- Выполните это задание, не подглядывая в книгу или свои записи.
- Завершив упражнение, сверьте результат с таблицами команд в концы глав и обратите внимание на первоначально забытые команды.
- Не волнуйтесь особенно о каждом пропущенном параметре или точности синтаксиса; достаточно записать лишь несколько первых слов команды.
- Делайте для последующего анализа примечания об абсолютно понятных командах и командах, в которых вы уверены меньше.
- Повторите это упражнение впоследствии, когда появится десять свободных минут, и сравните результат с примечаниями, сделанными в прошлый раз.

Ответы приведены в приложении П (Appendix O) на веб-сайте, но ваши диаграммы связей могут выглядеть иначе.

Диаграммы связей обзора части IV

Диаграмма	Описание	Где сохранен результат
1	Диаграмма связей команд	

В первых четырех частях этой книги было предоставлено достаточно информации для того, чтобы любой мог реализовать небольшую сеть IPv4. В следуюших двух частях тема IPv4 обсуждается подробней. В частности, часть V сосредоточивается скорее на концепциях IPv4 и лишь немного на конфигурации.

Во всех трех главах этой части обсуждается выбор варианта проекта сети IPv4, который может сделать инженер, и описывается значение этих вариантов. В частности, в главе 19 рассматривается проект подсети с единой маской, а в главе 20 речь пойдет об использовании нескольких масок подсети в одной классовой сети. Глава 21 завершает эту часть демонстрацией сокращения размера таблиц маршрутизации IP за счет суммирования маршрутов.

Часть V. Дополнительные концепции IPv4-адресации

ГЛАВА 19

Проект подсети

До сих пор в этой книге в основном обсуждались и использовались примеры IPv4 с заданными адресами и масками. В книге было уже представлено много примеров, но в них не требовалось выбирать IP-адрес или маску. Как упоминалось в главе 11, до сих пор в книге предполагалось, что некто уже разработал IP-адресацию и план подсетей, который осталось только реализовать.

Данная глава меняет эту модель. Она возвращается к последовательности проектирования и реализации сети IPv4, обсуждавшейся в главе 11 и показанной на рис. 19.1. Эта глава продолжает тему непосредственно после того, как некий сетевой инженер выбрал сеть класса A, B или C для корпоративной сети IPv4. Далее будет обсуждаться выбор проекта с единой маской для всех подсетей (первый раздел) и что означает выбор идентификатора подсети (второй раздел).

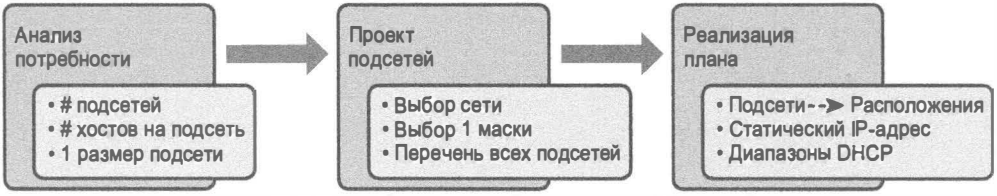


Рис. 19.1. Проект подсети и процесс его реализации из главы 11

В главе 20 рассматривается выбор другого проекта — масок подсети переменной длины (VLSM).

В этой главе рассматриваются следующие экзаменационные темы

IP-адресация (IPv4/IPv6)

Выбор подходящей схемы IPv4-адресации (использующей VLSM и суммирование), удовлетворяющей требования адресации в среде LAN/WAN.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Выбор маски, удовлетворяющей требованиям

В этом разделе рассматривается поиск всех масок, которые отвечают заданным требованиям по количеству подсетей и хостов на подсеть. Здесь подразумевается, что инженер уже определил эти требования и выбрал номер сети, которая будет разделена на подсети. Инженер также решил использовать одно значение маски во всей классовой сети.

Вооружившись информацией, изложенной в этой главе, можно отвечать на такие вопросы, как следующий, который имеет значение как для реальных технических задач, так и для экзаменов Cisco.

Вы используете сеть класса B 172.16.0.0. Необходимо 200 подсетей и 200 хостов на подсеть. Какие из следующих масок подсети отвечают требованиям? (Затем следует несколько возможных ответов с различными масками подсети.)

В начале текущего раздела приведен обзор концепций главы 11. В этом разделе представлены основные концепции того, как инженер при разработке соглашений подсети должен выбрать маску на основании требований.

Здесь, после общих концепций, связанных с темами главы 11, они рассматриваются глубже. В частности, рассматриваются три общих случая.

- Ни одна маска не отвечает требованиям.
- Одна и только одна маска отвечает требованиям.
- Требованиям отвечает несколько масок.

Для последнего случая рассматривается, как определить все маски, которые отвечают требованиям, и как выбрать из них наиболее подходящую.

Обзор: выбор минимального количества битов подсети и хоста

Сетевой инженер должен исследовать требования к количеству подсетей и хостов на подсеть, а затем выбрать маску. Как обсуждалось подробно в главе 13, классовое представление IP-адресов определяет структуру IP-адреса из трех частей: сети, подсети и хоста. Сетевой инженер должен выбрать маску так, чтобы количество битов подсети и хоста (S и H соответственно на рис. 19.2) отвечало требованиям.

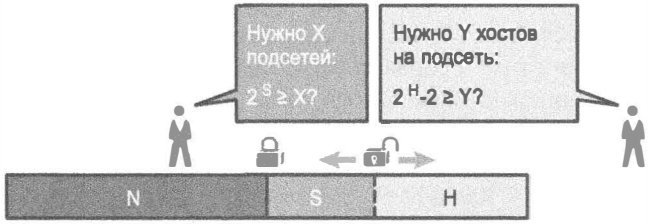


Рис. 19.2. Выбор количества битов подсети и хоста

В основном инженер должен выбрать биты подсети S так, чтобы количество подсетей, которое может быть уникально пронумеровано S битами (2^S), по крайней мере соответствовало количеству необходимых подсетей. Инженер применяет подобную логику к количеству битов хоста H, которое можно вычислить по формуле $2^H - 2$, поскольку два адреса в каждой подсети зарезервировано. Для удобства в табл. 19.1 приведены степени числа 2, она также будет полезна при решении этих задач.

Таблица 19.1. Степени числа 2. Справочник для разработки маски

Количество бит	2^X	Количество бит	2^X	Количество бит	2^X	Количество бит	2^X
1	2	5	32	9	512	13	8192
2	4	6	64	10	1024	14	16 384
3	8	7	128	11	2048	15	32 768
4	16	8	256	12	4096	16	65 536

Более формально процесс подразумевает определение минимальных значений S и H, которые отвечают требованиям. Ниже приведены начальные этапы выбора маски.

Этап 1 Определить количество битов сети (N) на основании ее класса

Этап 2 Определить наименьшее значение S по формуле $2^S => X$, где X — необходимое количество подсетей

Этап 3 Определить наименьшее значение H по формуле $2^H - 2 => Y$, где Y — необходимое количество хостов на подсеть

В следующих трех разделах исследуются эти начальные этапы выбора маски подсети.

Ни одна маска не отвечает требованиям

После определения необходимого количества битов подсети и хоста может оказаться, что они не вписываются в 32-разрядную маску подсети IPv4. Помните: маска всегда содержит в общей сложности 32 бита с двоичными единицами в части сети и подсети, а также двоичными нулями в части хоста. Условия экзаменационного вопроса могут содержать такие требования, для удовлетворения которых 32 битов не хватит.

Рассмотрим, например, следующий типичный экзаменационный вопрос.

Сетевой инженер создает проект подсети. Он планирует использовать сеть класса B 172.16.0.0. Есть потребность в 300 подсетях и 280 хостах на подсеть. Какую из следующих масок он мог бы выбрать?

Трехэтапный процесс, приведенный в предыдущем разделе, свидетельствует о том, что для удовлетворения этим требованиям понадобится в общей сложности 34 бита, поэтому не подойдет никакая маска. Поскольку используется сеть класса B, в адресе будет 16 битов сети и 16 битов хоста, — этого достаточно, чтобы создать части подсети и оставить биты хоста для каждой подсети. Количества битов подсети S=8 недостаточно, поскольку $2^8 = 256 < 300$, а S=9 достаточно, поскольку $2^9 = 512 => 300$. Аналогично, поскольку $2^8 - 2 = 254 < 280$, что меньше 300, 8 битов хоста недостаточно, а 9 битов ($2^9 - 2 = 510$) — вполне.

Эти требования не оставляют достаточно места для количества всех хостов и подсетей, так как части сети, подсети и хоста составляют в целом больше 32 битов.

- N=16. Поскольку сеть класса B, существует 16 битов сети.
- S=9 как минимум, поскольку 8 битов недостаточно для 300 подсетей ($2^8 = 256 < 300$), а 9 битов вполне достаточно ($2^9 = 512$).
- H=9 как минимум, поскольку 8 битов недостаточно для 280 хостов на подсеть ($2^8 - 2 = 254 < 280$), но 9 битов вполне достаточно ($2^9 - 2 = 510$).

На рис. 19.3 представлен полученный формат IP-адресов в этой подсети, после того, как инженер расписал 9 битов подсети на бумаге. Остается только 7 битов хоста, но инженер нуждается в 9.

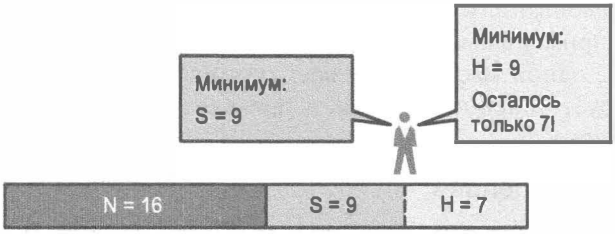


Рис. 19.3. Слишком мало битов для части хоста при данных требованиях

Требованиям отвечает только одна маска

Процесс, обсуждаемый в этой главе, частично сосредоточивается на поиске наименьшего количества битов подсети и хоста, удовлетворяющих требованиям. Если инженер попытается использовать эти минимальные значения и совместно биты частей сети, подсети и хоста составят в целом точно 32 бита, то требованиям отвечает только одна маска.

Рассмотрим, например, переделанную версию примера из предыдущего раздела с меньшими количествами подсетей и хостов.

Сетевой инженер создает проект подсети. Он планирует использовать сеть класса B 172.16.0.0. Есть потребность в 200 подсетях и 180 хостах на подсеть. Какую из следующих масок он мог бы выбрать?

Трехэтапный процесс определения минимального количества битов сети и хоста приводит к потребности в 16, 8 и 8 битах соответственно. Как и прежде, в сети класса B 16 битов сети. При потребности только в 200 хостах S=8 достаточно, поскольку $2^8 = 256 => 200$; 7 битов подсети недостаточно для 200 подсетей ($2^7 = 128$). Аналогично, поскольку $2^8 - 2 = 254 => 180$, 8 битов хоста отвечают требованиям; 7 битов хоста (для 126 хостов на подсеть) было бы недостаточно.

На рис. 19.4 приведен полученный формат IP-адресов в этой подсети.

На рис. 19.4 маска представлена концептуально. Для поиска фактического значения маски запишите ее в префиксном формате (/P), где P = N+S или, в данном случае, /24.

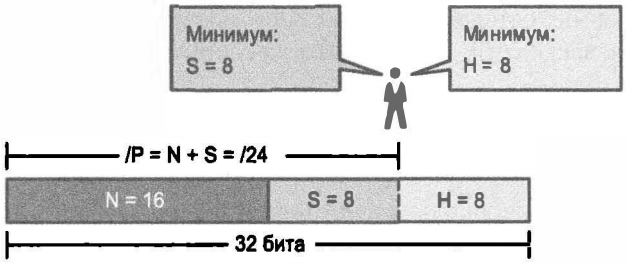


Рис. 19.4. Одна маска, отвечающая требованиям

Требованиям отвечает несколько масок

В зависимости от требований к количеству подсетей и хостов на подсеть, а также выбранной сети, требованиям могут отвечать несколько масок. В этих случаях необходимо найти все применимые маски. Затем появляется выбор, но что следует учитывать при выборе одной маски среди всех, которые отвечают требованиям? Этот раздел демонстрирует поиск всех масок, а также фактов, учитываемых при выборе одной маски из списка.

Поиск всех масок: концепции

Чтобы лучше разъяснить, как осуществляется поиск всех масок подсети в двоичном формате, в этом разделе описаны два главных этапа. На первом из них 32-разрядная двоичная маска подсети создается на бумаге. Записываются двоичные единицы для битов сети, двоичные единицы для битов подсети и двоичные нули для битов хоста, как всегда. Однако для S и H следует использовать минимальные значения. Когда все биты будут записаны, их может оказаться меньше 32!

Рассмотрим, например, следующую задачу, подобную приведенной ранее, но с некоторыми изменениями в требованиях.

Сетевой инженер создает проект подсети. Он планирует использовать сеть класса B 172.16.0.0. Есть потребность в 50 подсетях и 180 хостах на подсеть. Какую из следующих масок он мог бы выбрать?

Этот пример подобен прежнему, за исключением того, что в данном случае необходимо только 50 подсетей. Инженер снова использует частную сеть IP 172.16.0.0 с 16 битами сети. Проект в данном случае требует только 6 битов подсети, поскольку $2^6 = 64 => 50$, а также минимум 8 битов хоста.

Один из способов рассмотрения концепции поиска всех масок, которые отвечают этим требованиям, заключается в записи битов маски подсети: двоичные единицы для частей сети и подсети, а также двоичных нулей для части хоста. Однако считайте 32-разрядную маску 32-мя разрядными позициями, а двоичные нули *пишите только с правого края*. Общее представление приведено на рис. 19.5.

На рис. 19.5 приведено 30 битов, но у маски должно быть 32 бита. Два оставшихся бита могли бы стать битами подсети, будучи установленными в двоичные единицы. Эти же два бита могли бы быть битами хоста, будучи установленными в двоичные нули. Инженер просто должен решить, хочет ли он иметь больше битов подсети и больше подсетей или больше битов хоста для большего количества хостов на подсеть.

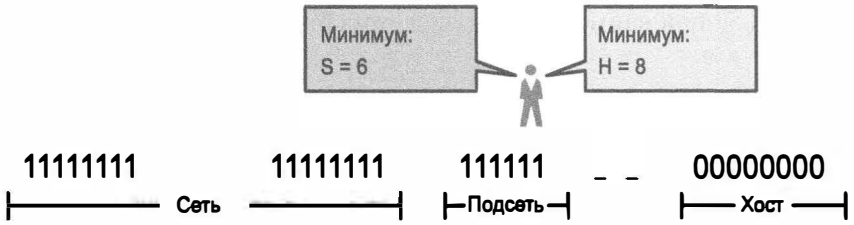


Рис. 19.5. Неполная маска N=16, S=6 и H=8

Но инженер не может выбрать любые значения для этих двух битов. Маска должна все еще выполнять следующее правило.

Факты о двоичных значениях в масках подсети

Ключевая тема

Маска подсети начинается со всех двоичных единиц и сопровождается всеми двоичными нулями без их чередования.

В примере с двумя не заполненными битами, представленном на рис. 19.5, одно значение (двоичное 01) нарушит это правило, а три другие комбинации двух битов (00, 10 и 11) — нет. В результате в этом примере требованиям отвечают три маски, как показано на рис. 19.6.

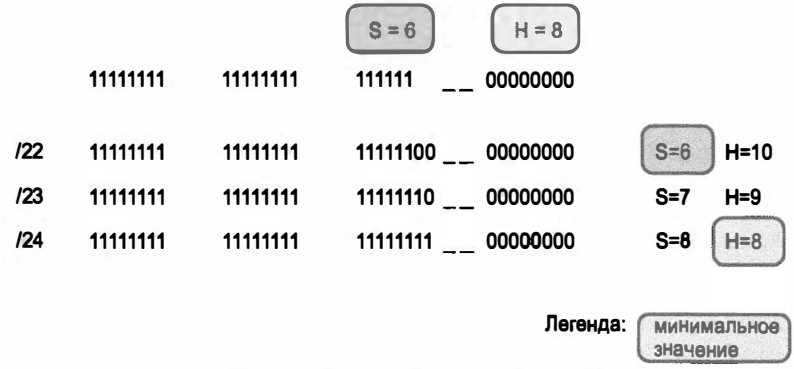


Рис. 19.6. Три маски, отвечающие требованиям

В трех масках первая имеет наименьшее количество битов подсети, а потому — больше битов хоста. Так, первая маска максимизирует количество хостов на подсеть. Последняя маска использует минимальное значение для количества битов хоста, позволяя использовать больше битов для подсети, продолжая все еще удовлетворять требования. В результате последняя маска максимизирует количество возможных подсетей.

Поиск всех масок: математика

Хотя концепции, связанные с примером, представленным на рис. 19.5 и 19.6, важны, диапазон отвечающих требованиям масок можно найти существенно легче, используя только простую математику. Как только стало известно значение N и минимальные значения S и H, процесс поиска маски требует лишь нескольких этапов.

Поиск значения /P при наименьших количествах битов подсети и хоста происходит следующим образом.

Ключевая тема **Более краткий процесс поиска всех префиксных масок, которые отвечают определенным требованиям**

Этап 1 Вычислите самую короткую префиксную маску (/P) на основании *минимального значения S*, где
$P = N + S$

Этап 2 Вычислите самую длинную префиксную маску (/P) на основании *минимального значения H*, где
$P = 32 - H$

Этап 3 Диапазон допустимых масок включает все значения /P между двумя значениями, вычисленными на предыдущих этапах

В примере, представленном на рис. 19.6, N = 16, минимум S = 6 и минимум H=8. На первом этапе выявляют самую короткую префиксную маску (/P с наименьшим значением) /22, при сложении N и S (16 + 6 = 22). Второй этап выявляет самую длинную отвечающую требованиям префиксную маску при вычитании наименьшего возможного значения для H (в данном случае 8) из числа 32. В результате получится маска /24. Третий этап напоминает, что диапазон от /22 до /24 включает /23, что также является возможным выбором.

Выбор лучшей маски

Если установленным требованиям отвечает несколько возможных масок, у инженера есть выбор. При этом, конечно, возникает вопрос: какую маску выбрать? Почему одна маска может быть лучше другой? Причины в итоге можно свести к трем основным пунктам.

Ключевая тема **Причины выбора одной маски подсети, а не другой**

- **Максимизировать количество хостов в подсети.** Чтобы сделать этот выбор, используйте самую короткую префиксную маску (т.е. маску с наименьшим значением /P), поскольку у нее наибольшая часть хоста.
- **Максимизировать количество подсетей.** Чтобы сделать этот выбор, используйте самую длинную префиксную маску (т.е. маску с наибольшим значением /P), поскольку у нее наибольшая часть подсети.
- **Увеличить количество и подсетей, и хостов.** Чтобы сделать этот выбор, используйте маску в середине диапазона, — это позволит предоставить больше битов и подсети, и хоста.

Например, на рис. 19.6 приведен диапазон масок /22–/24, отвечающих требованиям. У самой короткой маски, /22, наименьшее количество битов подсети и наибольшее количество битов хоста (10) из трех вариантов. Это максимизирует количество хостов в подсети. Самая длинная маска, /24, максимизирует количество битов подсети (8), увеличивая количество подсетей, по крайней мере в рамках соответствия исходным требованиям. Маска в середине, /23, предусматривает некий рост количества и подсетей, и хостов в подсети.

Формальный процесс

До сих пор в этой главе объяснялись различные этапы поиска масок подсети, удовлетворяющих требованиям проекта. Теперь они объединяются в общий список всего процесса, приведенный ниже. Обратите внимание, что в списке нет никаких новых концепций, которые не обсуждались бы ранее.

Полный процесс поиска и выбора масок, удовлетворяющих определенным требованиям

Ключевая тема

Этап 1 Найдите количество битов сети (N) согласно правилам класса

Этап 2 Вычислите минимальное количество битов подсети (S), чтобы 2^S было больше или равно количеству необходимых подсетей

Этап 3 Вычислите минимальное количество битов хоста (H), чтобы $2^H - 2$ было больше или равно количеству необходимых хостов в подсети

Этап 4 Если N+S+H > 32, то ни одна маска не удовлетворяет требованиям

Этап 5 Если N+S+H = 32, то требованиям удовлетворяет только одна маска. Вычислите маску как /P, где P = N+S

Этап 6 Если N+S+H < 32, то требованиям удовлетворяет несколько масок.

А. Вычислите маску /P на основании минимального значения S, где P = N+S. Эта маска максимизирует количество хостов в подсети.

Б. Вычислите маску /P на основании минимального значения H, где P = 32 - H. Эта маска максимизирует количество возможных подсетей.

В. Получите полный диапазон масок, включив все префиксные длины между двумя значениями, вычисленными на этапах А и Б.

Практические задания

Прежде чем перейти к следующему разделу, попрактикуйтесь в процессах, обсуждаемых в данной главе, пока не станете почти всегда получать правильные ответы. Используйте любые средства по своему усмотрению и любое время. Затем можно продолжить чтение.

Перед сдачей экзамена потренируйтесь, чтобы овладеть темой данной главы полностью и отвечать достаточно быстро. Что касается времени, то ответ (все маски, отвечающие требованиям, которые максимизируют количество подсетей или хостов) нужно давать приблизительно через 15 секунд. Ключевые концепции и рекомендации такого двухэтапного подхода приведены в табл. 19.2.

Таблица 19.2. Продолжайте читать с учетом целей экзамена по темам данной главы

Период	Перед переходом к следующей главе	Перед сдачей экзамена
Сосредоточиться на ...	теме изучения	Быть быстрым и правильным
Разрешенные средства	Все	Ваш мозг и блокнот
Цель: точность	90% правильных ответов	100% правильных ответов
Цель: скорость	Любая скорость	15 секунд

Практические задания по выбору маски подсети

Ниже приведены три отдельных задания с адресом классовой сети, а также необходимым количеством подсетей и хостов на подсеть. Для каждого задания определите минимальное количество битов подсети и хоста, которые отвечают требованиям. Если возможно несколько масок, обратите внимание на то, какая маска максимизирует количество хостов в подсети, а какая количество подсетей. Если требованиям отвечает только одна маска, укажите ее. Приведите маски в префиксном формате.

1. Сети 10.0.0.0 требуется 1500 подсетей и 300 хостов на подсеть.
2. Сети 172.25.0.0 требуется 130 подсетей и 127 хостов на подсеть.
3. Сети 192.168.83.0 требуется 8 подсетей и 8 хостов на подсеть.

Ответы приведены в конце главы.

Дополнительные практические задания

Ниже перечислены некоторые возможности для дополнительной практики.

- Приложение Ж, содержащее дополнительные практические задания, а также объяснения по поиску ответа к каждому заданию.
- Создайте собственные задания. Большинство калькуляторов подсети позволяют ввести класс A, B или C сети и выбрать маску, затем калькулятор перечислит количество подсетей и хостов в подсети, созданных этой сетью и маской. Выберите номер сети и необходимые количества подсетей и хостов, получите ответы и проверьте результат с калькулятором.

Поиск всех идентификаторов подсети

После создания плана подсетей IP с выбором единой маски для всей сети класса A, B или C следует присвоить уникальные *идентификаторы подсети* (subnet ID) для каждой конкретной сети VLAN, последовательного канала связи и других элементов объединенной сети, нуждающихся в подсетях. Но что такое идентификатор подсети? Кроме того, как после выбора идентификатора сети и единой маски для всех подсетей найти все идентификаторы подсетей, используя только несложную математику. Данный раздел посвящен этой математике и единственному вопросу:

Дана одна сеть класса A, B или C и одна маска подсети, используемая для всех подсетей. Каковы все идентификаторы подсети?

Чтобы получить ответ на этот вопрос, можно решить проблему двоичным или десятичным способом. В этой главе используется десятичный подход. Хотя сам процесс требует лишь простой математики, большинству людей потребуются практические навыки, чтобы быстро и уверенно отвечать на этот вопрос.

Десятичный процесс начинается с выявления первого, или самого младшего, идентификатора подсети. Затем процесс выявляет шаблон всех идентификаторов подсети для данной маски подсети, чтобы можно было найти каждый последующий идентификатор подсети с помощью простого сложения. Сначала в этом разделе рассматриваются ключевые идеи данного процесса, а затем дано формальное определение процесса.

ВНИМАНИЕ!

В некоторых видеофильмах на образе DVD-диска, находящегося на веб-странице книги, продемонстрированы те же фундаментальные процессы поиска всех идентификаторов подсети. Вы можете просмотреть их до или после чтения данного раздела или вместо того, чтобы читать его, пока не узнаете, как независимо найти все идентификаторы подсети. Нумерация этапов процесса на видео может не соответствовать этапам, изложенным в данном издании книги.

Первый идентификатор подсети: нулевая подсеть

Первый этап поиска всех идентификаторов подсети одной сети невероятно прост: скопируйте идентификатор сети. Например, возьмите идентификатор сети класса A, B или C, другими словами, идентификатор классовой сети, и запишите его как первый идентификатор подсети. Независимо от класса (A, B или C) используемой сети и маски подсети, первый идентификатор подсети совпадает с идентификатором сети.

Например, если речь идет о классовой сети 172.20.0.0, то независимо от маски первым идентификатором подсети будет 172.20.0.0.

Этот первый идентификатор подсети в каждой сети имеет два специальных названия: *нулевая подсеть* (zero subnet) или *подсеть нуль* (subnet zero). Первоначально происхождение этих названий связывали с тем фактом, что у нулевой подсети сети в двоичном представлении вся часть подсети занята двоичными нулями. В десятичном представлении нулевая подсеть может быть легко выявлена по тому, что в числовом виде она всегда совпадает с идентификатором самой сети.

ВНИМАНИЕ!

В последнее время в проектах подсетей IP, как правило, не используют нулевую подсеть во избежание недоразумений, которые могут возникнуть из-за совпадения идентификатора сети с идентификатором подсети.

Поиск шаблона с использованием магического числа

Идентификаторы подсети следуют вполне предсказуемому шаблону, по крайней мере, в случае использования единой маски для всех подсетей в сети. Шаблон использует *магическое число* (magic number), обсуждавшееся в главе 14. Напомним: магическое число — это 256 минус десятичное значение маски в определенном октете, который в данной книге именуется *интересующим октетом* (interesting octet).

На рис. 19.7 приведены четыре примера шаблонов с четырьмя разными масками. Начнем с примера вверху рисунка. Слева указана маска 255.255.128.0. Интересующий октет — третий, его значение не 0 и не 255. Слева указано также магическое число, вычисленное как 256 – 128 = 128. Шаблон идентификаторов подсети представлен на числовой оси; т.е. у идентификаторов подсети при использовании этой маски в третьем октете будет 0 или 128. Например, при использовании сети 172.16.0.0 идентификаторами подсетей будут 172.16.0.0 и 172.16.128.0.

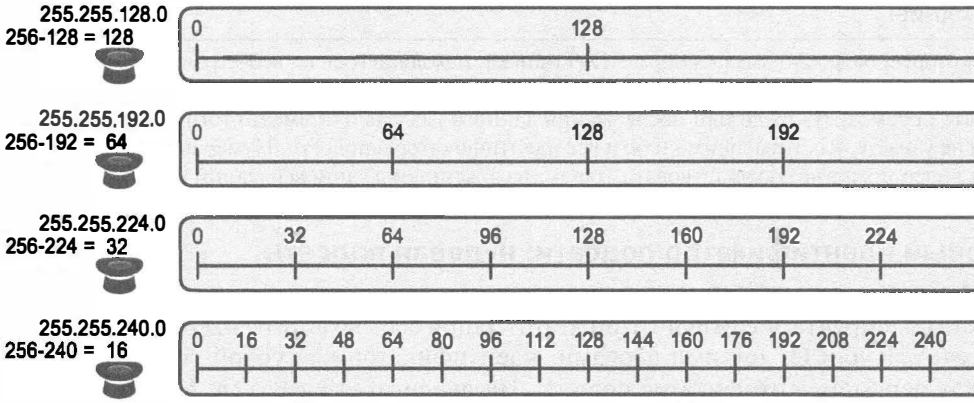


Рис. 19.7. Шаблоны с магическими числами для масок /17 – /20

Теперь сосредоточьтесь на втором примере с маской 255.255.192.0. Магическое число здесь 64 (256 – 192 = 64), поэтому идентификаторы подсети в третьем октете будут кратны 64, т.е. 0, 64, 128, или 192. Например, при использовании с сетью 172.16.0.0 идентификаторами подсетей были бы 172.16.0.0, 172.16.64.0, 172.16.128.0 и 172.16.192.0.

В третьем примере представлена маска 255.255.224.0 с магическим числом 256 – – 224 = 32. Как можно заметить в центре рисунка, значения идентификаторов подсети будут кратны 32. Например, при использовании снова с сетью 172.16.0.0 эта маска дала бы идентификаторы подсетей 172.16.0.0, 172.16.32.0, 172.16.64.0, 172.16.96.0 и т.д.

И наконец, пример внизу имеет маску 255.255.240.0 и магическое число 16 в третьем октете. Поэтому все идентификаторы подсети будут кратны 16 в третьем октете (их значения представлены в средней части рисунка).

Формальный процесс с менее чем 8 битами подсети

Хотя шаблоны на рис. 19.7 вполне очевидны, может быть не до конца понятно, как применить эти концепции для поиска всех идентификаторов подсети в любом случае. Этот раздел посвящен исключительно процессу поиска всех идентификаторов подсети.

Чтобы упростить объяснения, в этом разделе подразумевается, что битов подсети меньше 8. Далее, в разделе "Поиск всех подсетей с более чем 8 битами подсети", описан полный процесс, применимый во всех случаях.

Вначале, чтобы упорядочить свои мысли, возможно, имеет смысл свести данные в таблицу, подобную табл. 19.3. В книге такая таблица называется обобщенным перечнем всех подсетей.

Таблица 19.3. Обобщенный перечень всех подсетей

Октет	1	2	3	4
Маска				
Магическое число				
Номер сети/нулевая подсеть				
Первая не нулевая подсеть				

Окончание табл. 19.3

Октет	1	2	3	4
Следующая подсеть				
Последняя подсеть				
Широковещательная подсеть				
Вне диапазона — используется процессом				

Формальный процесс поиска всех идентификаторов подсети, исходя из номера сети и единой маски подсети, имеет следующий вид.

Этапы формального процесса поиска всех идентификаторов подсети, когда битов подсети меньше 8

Ключевая тема

Этап 1 В первом пустом ряду таблицы запишите маску подсети в десятичном формате

Этап 2 Выявите интересующий октет, который является единственным октетом маски со значением не 255 и не 0. Нарисуйте прямоугольник вокруг столбца интересующего октета

Этап 3 Вычислите и запишите магическое число при вычитании значения интересующего октета маски подсети из 256

Этап 4 В следующем пустом ряду перечня всех подсетей запишите номер классовой сети, совпадающий с номером нулевой подсети

Этап 5 Для поиска каждого последующего номера подсети:

А. Для трех не интересующих октетов скопируйте значения предыдущего номера подсети.

Б. Для интересующего октета добавьте магическое число к интересующему октету предыдущего номера подсети

Этап 6 Как только сумма, вычисленная на этапе 5 Б, будет равна 256, остановите процесс. Число 256 — вне диапазона, а предыдущий номер подсети — широковещательный адрес подсети

Хотя описание способа очень длинное, обладая практическим навыком, большинство людей смогут найти так ответы намного быстрее, чем при использовании двоичной математики. Как обычно, изучать этот процесс лучше на практике, а не только теоретически. Для этого выполните задания, просмотрите видеофильмы, находящиеся на образе DVD-диска, а также рассмотрите дополнительные примеры на нем.

Пример 1: сеть 172.16.0.0, маска 255.255.240.0

Вначале сосредоточимся на первых четырех из шести этапов, когда сеть 172.16.0.0 разделена на подсети маской 255.255.240.0. Результат этих первых четырех этапов приведен на рис. 19.8.

Этап 1 Записываем маску 255.255.240.0, которая была дана в условии задачи. (На рис. 19.8 для справки приведен также идентификатор сети 172.16.0.0.)

Этап 2 Третий октет маски не 0 и не 255, что делает его интересующим

Этап 3 Поскольку значение маски в третьем октете 240, магическое число равно 256 – 240 = 16

Этап 4 Поскольку идентификатор сети 172.16.0.0, первый идентификатор подсети (нулевая подсеть) также 172.16.0.0.

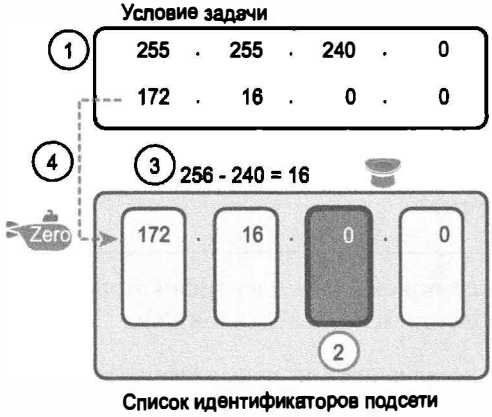


Рис. 19.8. Результат выполнения первых четырех этапов: 172.16.0.0, 255.255.240.0

На этих четырех первых этапах выясняется первая подсеть (нулевая), что позволяет осуществить остальные этапы, выявив интересующий октет и магическое числа. Пятый этап процесса требует скопировать значения трех не интересующих октетов и добавить магическое число (в данном случае 16) к интересующему октету (в данном случае к октету 3). Повторяйте этот этап, пока значение интересующего октета не станет равно 256 (этап 6). Достигнув числа 256, вы имеете все идентификаторы подсети, а значение 256 вычеркните, так как оно является не корректным идентификатором подсети. Результат выполнения этих этапов приведен на рис. 19.9.

Ключевая тема

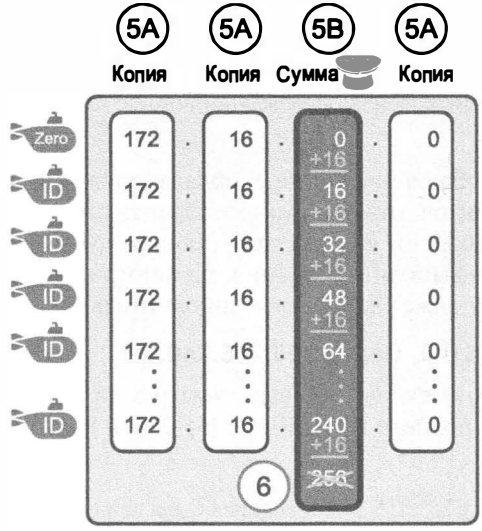


Рис. 19.9. Пример добавления магического числа к интересующему октету при поиске всех идентификаторов подсети

ВНИМАНИЕ!

В любом списке всех идентификаторов подсети самый верхний ее идентификатор называется *широковещательной подсетью* (broadcast subnet). Несколько десятилетий назад инженеры избегали использования широковещательной подсети. Однако ее использование не вызывает проблем. Термин *широковещательная подсеть* возник благодаря тому факту, что широковещательный адрес подсети в широковещательной подсети совпадает с общесетевым широковещательным адресом.

ВНИМАНИЕ!

Термины *широковещательная подсеть* и *широковещательный адрес подсети* иногда путают. *Широковещательная подсеть* (broadcast subnet) — это одна из подсетей, а именно самая старшая подсеть; в сети существует только одна такая подсеть. Термин *широковещательный адрес подсети* описывает один номер в каждой подсети, который в цифровом виде является самым старшим номером в этой подсети.

Пример 2: сеть 192.168.1.0, маска 255.255.255.224

В сети класса C с маской 255.255.255.224 интересующим октетом в данном примере является четвертый октет. Однако процесс тот же, с той же логикой, только применяется он к другому октету. Как и в предыдущем примере, следующий список содержит первые четыре этапа, а рис. 19.10 демонстрирует их результаты.

Этап 1 Записываем маску 255.255.255.224, которая была дана в условии задачи, и опционально записываем номер сети (192.168.1.0)

Этап 2 Четвертый октет маски не 0 и не 255, что делает его интересующим

Этап 3 Поскольку значение маски в четвертом октете 224, магическое число равно 256 – 224 = 32

Этап 4 Поскольку идентификатор сети 192.168.1.0, первый идентификатор подсети (нулевая подсеть) также 192.168.1.0

Рис. 19.10. Результаты первых четырех этапов: 192.168.1.0, 255.255.255.224

Пятый этап процесса требует скопировать значения первых трех октетов, а затем добавить магическое число (в данном случае 32) к интересующему октету (в данном случае к октету 4). Повторяйте этот этап, пока значение интересующего октета не

станет равно 256 (этап 6). Достигнув числа 256, вы имеете все идентификаторы подсети, а значение 256 вычеркните, так как оно является не корректным идентификатором подсети. Результат этих этапов приведен на рис. 19.11.

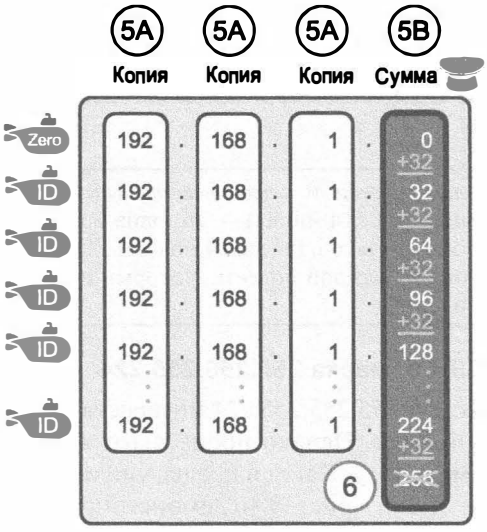


Рис. 19.11. Идентификаторы подсети: 192.168.1.0, 255.255.255.224

Поиск всех подсетей точно с 8 битами подсети

Формальный процесс в предыдущем разделе определял интересующий октет как октет, значение маски которого не равнялось ни 255, ни 0. Если маска определяет ровно 8 битов подсети, следует использовать совершенно другую логику выявления интересующего октета; в противном случае применяется тот же процесс. Фактически реальные идентификаторы подсети могут быть немного более интуитивно понятными.

Существуют только два случая с точно 8 битами подсети.

- Сеть класса A с маской 255.255.0.0; весь второй октет содержит биты подсети.
- Сеть класса B с маской 255.255.255.0; весь третий октет содержит биты подсети.

В любом случае используйте тот же процесс, что и с менее чем 8 битами подсети, но определите интересующий октет как содержащий биты подсети. Кроме того, поскольку значение маски 255, магическое число будет 256 – 255 = 1, таким образом, последующий идентификатор подсети будет на единицу больше предыдущего.

Например, для 172.16.0.0 и маски 255.255.255.0 интересующий октет третий, а магическое число 256 – 255 = 1. Начиная с нулевой подсети, равной по значению номеру сети 172.16.0.0, добавляйте далее 1 в третьем октете. Например, первые четыре подсети следующие:

- 172.16.0.0 (нулевая подсеть);
- 172.16.1.0;
- 172.16.2.0;
- 172.16.3.0.

Поиск всех подсетей с более чем 8 битами подсети

Выше, в разделе "Формальный процесс с менее чем 8 битами подсети", для упрощения изучения подразумевалось наличие менее 8 битов подсети. В реальной жизни необходимо уметь найти все идентификаторы подсети при любой допустимой маске, поэтому нельзя полагать, что битов подсети окажется меньше 8.

В примерах, где битов подсети по крайней мере 9, есть как минимум 512 идентификаторов подсети, поэтому запись такого списка заняла бы слишком много времени. Для экономии места в примерах будут использоваться сокращения вместо перечисления сотен или тысяч идентификаторов подсети.

Процесс с менее чем 8 идентификаторами подсети подразумевает инкремент магического числа в одном октете. При более чем 8 битах подсети новый дополненный процесс подразумевает приращение в нескольких октетах. Поэтому в данном разделе описаны два общих случая: когда существует 9–16 битов подсети, подразумевающий, что поле подсети составляет только два октета; и случай с 17 или более битами подсети, подразумевающий поле подсети в трех октетах.

Процесс с 9–16 битами подсети

Чтобы понять процесс, необходимо ознакомиться с несколькими используемыми в нем терминами. На рис. 19.12 приведены подробности концепции на примере использования сети класса B 130.4.0.0 и маски 255.255.255.192. В нижней части приведена структура адресов по маске: часть сети из двух октетов, поскольку это адрес класса B, часть подсети на 10 битов по маске (/26) и 6 битов хоста.

Рис. 19.12. Фундаментальные концепции и термины процесса с более чем 8 битами подсети

В данном случае биты подсети расположены в двух октетах: 3 и 4. Крайний правый из этих октетов — интересующий октет, а октет слева условно именуется *просто левым* (just-left) октетом.

Видоизмененный процесс предполагает приращение магического числа в интересующем октете и приращение на единицу в просто левом октете.

Формальные этапы поиска всех идентификаторов подсети, когда существует больше 8 битов подсети

Ключевая тема

Этап 1 Вычислите идентификаторы подсети, используя процесс для 8 битов подсети или меньше. Когда сумма дойдет до значения 256, переходите к следующему этапу; считайте перечисленные идентификаторы подсети *блоком подсетей* (subnet block)

Этап 2 Скопируйте предыдущий блок подсетей, но добавьте 1 к просто левому октету во всех идентификаторах подсети нового блока

Этап 3 Повторяйте этап 2 до тех пор, пока не создадите блок с просто левым октетом 255, а затем удалите его

Честно говоря, формальная концепция может вызывать проблемы, если не попрактиковаться на примерах. Если процесс остается немного неясным, лучше рассмотрите следующие примеры вместо повторного чтения формального процесса.

Сначала рассмотрим пример на основании рис. 19.12, с сетью 130.4.0.0 и маской 255.255.255.192. Структура уже представлена на рис. 19.12, а на рис. 19.13 приведен блок идентификаторов подсети, созданный на этапе 1.

Рис. 19.13. Этап 1: список первого блока идентификаторов подсети

Логика на этапе 1, подразумевающая создание блока из четырех идентификаторов подсети, следует тому же процессу с магическим числом, что и прежде. Первый идентификатор подсети, 130.4.0.0, является нулевой подсетью. Каждый из следующих трех идентификаторов подсети на 64 больше, поскольку магическое число в данном случае 256 – 192 = 64.

Этапы 2 и 3 формального процесса демонстрируют, как создать 256 блоков подсетей. Сделав это, можно перечислить все 1024 идентификатора подсети. Для этого создайте 256 полных блока подсети: один с 0 в просто левом октете, один с 1 в просто левом октете, далее с 2 в просто левом октете, и так далее до 255. Процесс продолжается до тех пор, пока не будет создан блок подсетей со значением 255 в просто левом октете (в данном случае в третьем октете). На рис. 19.14 представлена концепция добавления в первых нескольких блоках подсетей.

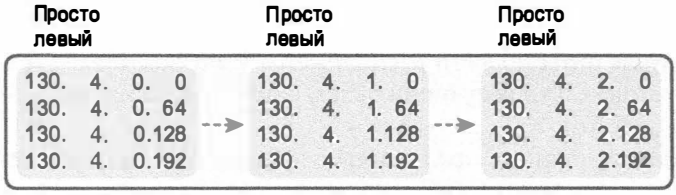


Рис. 19.14. Этап 2: репликация блока подсетей с добавлением единицы в просто левом октете

Этот пример с 10 полными битами подсети создает 256 блоков по 4 подсети в каждом для 1024 подсетей в общей сложности. Эта математика соответствует обычному методу подсчета подсетей, поскольку $2^{10} = 1024$.

Процесс с 17 и более битами подсети

Чтобы создать проект подсети, допускающей 17 и более битов подсети, следует использовать сеть класса A. Кроме того, часть подсети будет состоять из второго и третьего октетов полностью плюс часть четвертого октета. Это означает множест-

во идентификаторов подсети: по крайней мере, 2^{17} (или 131 072) подсети. На рис. 19.15 приведен пример именно такой структуры с сетью класса A и маской /26.

Рис. 19.15. Структура адреса с 18 битами подсети

Для поиска всех идентификаторов подсети в этом примере используйте тот же общий процесс, что и с 9–16 битами подсети, но с большим количеством блоков подсетей. В действительности вы должны создать блоки подсетей для всех комбинаций значений (0–255 включительно) и во втором, и в третьем октете. Общее представление приведено на рис. 19.16. Обратите внимание: только при 2 битах подсети в четвертом октете в этом примере будет по четыре подсети в каждом блоке подсетей.

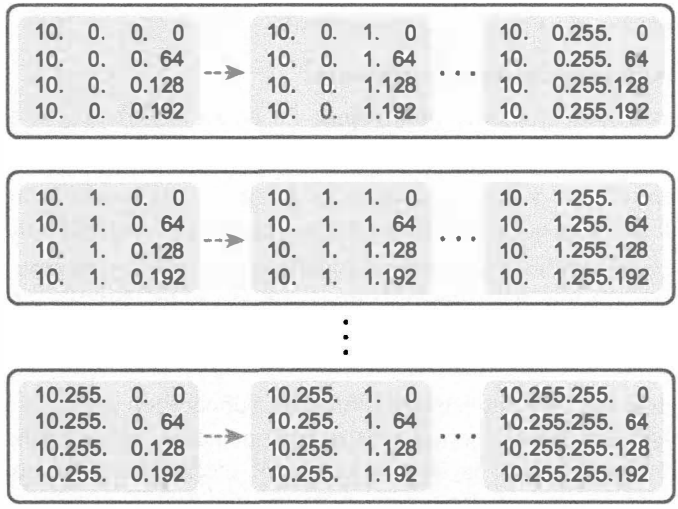


Рис. 19.16. 256 групп по 256 блоков из четырех подсетей

Практика поиска всех идентификаторов подсети

Прежде чем перейти к следующей главе, попрактикуйтесь в процессах, обсуждаемых в данной главе, пока не станете получать правильные ответы почти всегда. Используйте любые средства по своему усмотрению и любое время. Затем можно продолжить чтение.

Перед сдачей экзамена потренируйтесь, чтобы овладеть темой данной главы полностью и отвечать достаточно быстро. Достижение разумной скорости решения задачи довольно трудно, поскольку одни комбинации идентификатора сети и маски могут давать сотни или тысячи подсетей, в то время как другие — значительно меньшие количества. Поэтому перед экзаменом следует быть в состоянии выявить первые четыре подсети, которые включают нулевую подсеть, за 45 секунд. Ключевые концепции и рекомендации такого двухэтапного подхода приведены в табл. 19.4.

Таблица 19.4. Продолжайте читать с учетом целей экзамена по темам данной главы

Период	Перед переходом к следующей главе	Перед сдачей экзамена
Сосредоточиться на ...	теме изучения	быть быстрым и правильным
Разрешенные средства	Все	Ваш мозг и блокнот
Цель: точность	90% правильных ответов	100% правильных ответов
Цель: скорость	Любая скорость	45 секунд

Практические задания по поиску всех идентификаторов подсети

Вот список из трех заданий, где даны номер классовой сети и маска в префиксном стиле. Найдите все идентификаторы подсети для каждой задачи.

1. 192.168.9.0/27
2. 172.30.0.0/20
3. 10.0.0.0/17

Ответы даны ниже, в разделе "Ответы на приведенные ранее практические задания по поиску всех идентификаторов подсети".

Дополнительные практические задания

В этом разделе перечислены некоторые из возможностей для дополнительной практики.

- Приложение Ж, в котором содержатся дополнительные практические задания, а также даны объяснения по поиску ответа к каждому заданию.
- Создайте собственные задания. Большинство калькуляторов подсети позволяют найти все идентификаторы подсети, когда вводят IP-адрес и маску. Поэтому запишите идентификатор сети и маску на бумаге, найдите ответ, а затем введите значения в калькулятор, чтобы проверить результат.
- Приложение Find All Subnets для iPhone на Subnet Prep (`www.subnetprep.com`) предоставляет обзорное видео, но что важнее всего, практические задания. Как обычно, оно позволяет изучить описанный здесь процесс и попрактиковаться, причем не привязываясь к какому-либо специфическому процессу.
- Просмотрите видеофильмы на образе DVD-диска, демонстрирующие процессы, описанные в этой главе.

Обзор

Резюме

- Сетевой инженер должен исследовать требования к количеству подсетей и хостов на подсеть, а затем выбрать маску. Классовое представление IP-адресов определяет структуру IP-адреса из трех частей: сети, подсети и хоста. Сетевой инженер должен выбрать маску так, чтобы количество битов подсети и хоста отвечало требованиям.
- В проекте подсети есть три возможности.
  - Ни одна маска не отвечает требованиям.
  - Одна и только одна маска отвечает требованиям.
  - Требованиям отвечает несколько масок.
- Если установленным требованиям отвечает несколько возможных масок, у инженера есть выбор. При этом, конечно, возникает вопрос: какую маску выбрать? Почему одна маска может быть лучше другой? Причины в итоге можно свести к трем основным пунктам.
  - **Максимизировать количество хостов в подсети**. Чтобы сделать этот выбор, используйте самую короткую префиксную маску (т.е. маску с наименьшим значением /P), поскольку у нее наибольшая часть хоста.
  - **Максимизировать количество подсетей**. Чтобы сделать этот выбор, используйте самую длинную префиксную маску (т.е. маску с наибольшим значением /P), поскольку у нее наибольшая часть подсети.
  - **Увеличить количество и подсетей, и хостов**. Чтобы сделать этот выбор, используйте маску в середине диапазона, — это позволит предоставить больше битов и подсети, и хоста.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Проект создания подсетей IP компании находится на стадии реализации. В настоящий момент главный инженер решил использовать сеть класса B 172.23.0.0. Проект требует 100 подсетей с наибольшей подсетью, нуждающейся в 500 хостах. Руководство требует, чтобы проект предусматривал 50-процентный рост количества подсетей и размера наибольшей подсети. Требуется также использовать единую маску по всей сети класса B. Сколько масок отвечает этим требованиям?

   А) 0

   Б) 1

   В) 2

   Г) 3+

2. Проект подсетей IP требует 200 подсетей, 120 хостов на подсеть в наибольшей подсети и использования единой маски в одной частной сети IP. Проект требует также запланировать 20-процентный рост количества подсетей и хостов в наибольшей подсети. Какие из следующих ответов определяют частную сеть IP и маску, отвечающую этим требованиям?

   А) 10.0.0.0/25

   Б) 10.0.0.0/22

   В) 172.16.0.0/23

   Г) 192.168.7.0/24

3. Инженер планирует использовать сеть класса B 172.19.0.0 и единую маску подсети по всей сети. В ответах перечислены маски, которые рассматривает инженер. Выберите из них маску, которая обеспечит наибольшее количество хостов на подсеть при достаточном количестве битов для поддержки 1000 подсетей.

   А) 255.255.255.0

   Б) /26

   В) 255.255.252.0

   Г) /28

4. Проект подсети использует сеть класса A 10.0.0.0, и инженер должен выбрать единую маску для всей сети. Проект требует 1000 подсетей с наибольшей подсетью в 200 хостов. Какая из следующих масок отвечает требованиям при максимальном количестве подсетей?

   А) /18

   Б) /20

   В) /22

   Г) /24

5. Инженер создал последовательный список идентификаторов подсети для сети 172.30.0.0/22. Подразумевается, что маска /22 используется по всей сети. Какое из следующих утверждений истинно? (Выберите два ответа.)

   А) Любые два последовательных идентификатора подсети отличаются значением 22 в третьем октете.

   Б) Любые два последовательных идентификатора подсети отличаются значением 16 в четвертом октете.

   В) Список содержит 64 идентификатора подсети.

   Г) Последний идентификатор подсети 172.30.252.0.

6. Какой из приведенных ниже идентификаторов подсети допустим для сети 192.168.9.0 при использовании маски /29 с учетом, что она используется по всей сети?

   А) 192.168.9.144

   Б) 192.168.9.58

   В) 192.168.9.242

   Г) 192.168.9.9

7. Какой из перечисленных ниже идентификаторов подсети не допустим для сети 172.19.0.0 при использовании единой для всей сети маски /24?

   А) 172.19.0.0

   Б) 172.19.1.0

   В) 172.19.255.0

   Г) 172.19.0.16

8. Какой из перечисленных ниже идентификаторов подсети не допустим для сети 10.0.0.0 при использовании единой для всей сети маски /25?

   А) 10.0.0.0

   Б) 10.255.255.0

   В) 10.255.127.128

   Г) 10.1.1.192

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 19.5.

Таблица 19.5. Ключевые темы главы 19

Элемент	Описание	Страница
Факты	Факты о двоичных значениях в масках подсети	583
Список	Более краткий процесс поиска всех префиксных масок, которые отвечают определенным требованиям	584
Список	Причины выбора одной маски подсети, а не другой	584
Список	Полный процесс поиска и выбора масок, удовлетворяющих определенным требованиям	585
Список	Этапы формального процесса поиска всех идентификаторов подсети, когда битов подсети меньше 8	589
Рис. 19.9	Пример добавления магического числа к интересующему октету при поиске всех идентификаторов подсети	590
Список	Формальные этапы поиска всех идентификаторов подсети, когда существует больше 8 битов подсети	593

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

нулевая подсеть (zero subnet), подсеть нуль (subnet zero), широковещательная подсеть (broadcast subnet)

Практика

Если это еще не сделано, попрактикуйтесь в вопросах поиска всех масок подсети на основании требований, описанных в данной главе. Рекомендации приведены в разделе "Практические задания".

Ответы на приведенные ранее практические задания по выбору маски подсети

В разделе "Практические задания по выбору маски подсети" приведены три практических задания. Ответы приведены в табл. 19.6. После таблицы следуют примечания к каждому заданию.

Таблица 19.6. Практическое задание: поиск масок, которые отвечают требованиям

Задание	Класс	Минимум битов подсети	Минимум битов хоста	Префиксный диапазон	Префикс для максимизации подсетей	Префикс для максимизации хостов
1	A	11	9	/19 – /23	/23	/19
2	B	8	8	/16	Нет	Нет
3	C	3	4	/27 – /28	/28	/27

1. N=8, поскольку в задаче упоминается сеть класса A 10.0.0.0. С потребностью в 1500 подсетях 10 битов подсети обеспечивают только 1024 подсети (согласно табл. 19.1), а 11 битов подсети (S) предусмотрели бы 2048 подсетей, что больше необходимых 1500. Аналогично наименьшим количеством битов хоста будет 9, поскольку 2^{*} – 2 = 254, а проект требует 300 хостов на подсеть. Самая короткая префиксная маска была бы /19, вычисляемая при добавлении N (8) к наименьшему допустимому для использования количеству битов подсети S (11). Аналогично с минимальным значением H (9) самая длинная префиксная маска максимизирует количество подсетей, 32 – H = /23.
2. N=16, поскольку в задаче упоминается сеть класса B 172.25.0.0. С потребностью в 130 подсетях 7 битов подсети обеспечивают только 128 подсетей (согласно табл. 19.1), а 8 битов подсети (S) предусмотрели бы 256 подсетей, что больше необходимых 130. Аналогично наименьшим количеством битов хоста будет 8, поскольку 2^{7} – 2 = 126 близко к необходимым 127, но не совсем достаточно, что делает H = 8 наименьшим количеством битов хоста, которое отвечает требованиям. Обратите внимание: в сумме количество битов сети, минимальное количество битов подсети и хоста составит 32, таким образом, только одна маска отвечает требованиям, а именно /24, вычисляемая при суммировании количества битов сети (16) и подсети (8).
3. N=24, поскольку в задаче упоминается сеть класса C 192.168.83.0. С потребностью в 8 подсетях 3 бита подсети обеспечивают их недостаточно. Наименьшее количество битов хоста будет 4, поскольку 2^{3} – 2 = 6, а проект требует 8 хостов на подсеть. Самая короткая префиксная маска была бы /27, она вычисляется при добавлении N (24) к наименьшему пригодному для использования количеству битов подсети S (3). Аналогично при минимальном значении H (4) самая длинная префиксная маска максимизирует количество подсетей, 32 – H = /28.

Ответы на приведенные ранее практические задания по поиску всех идентификаторов подсети

В разделе "Практические задания по поиску всех идентификаторов подсети" приведены три практических задания. Ответы даны в табл. 19.7-19.8. После таблиц следуют примечания к каждому заданию.

Ответ на практическое задание 1

В первом задании даны сеть 192.168.9.0 и маска /27. Маска преобразуется в маску DDN 255.255.255.224. При использовании сети класса C имеется 24 бита сети и только 3 бита подсети, находящихся в четвертом октете. Таким образом, это случай с менее чем 8 битами подсети и четвертым интересующим октетом.

Для начала запишите нулевую подсеть, а затем начинайте добавлять магическое число в интересующий октет. Нулевая подсеть равна идентификатору сети (в данном случае 192.168.9.0). Магическое число, вычисленное как 256 – 224 = 32, следует добавить к предыдущему интересующему октету идентификатора подсети. Результаты см. в табл. 19.7.

Таблица 19.7. Перечень всех подсетей: 192.168.9.0/27

Октет	1	2	3	4
Маска	255	255	255	224
Магическое число				32
Номер сети/нулевая подсеть	192	168	9	0
Первая не нулевая подсеть	192	168	9	32
Следующая подсеть	192	168	9	64
Следующая подсеть	192	168	9	96
Следующая подсеть	192	168	9	128
Следующая подсеть	192	168	9	160
Следующая подсеть	192	168	9	192
Широковещательная подсеть	192	168	9	224
Вне диапазона — используется процессом	192	168	9	256

Ответ на практическое задание 2

Во втором задании даны сеть 172.30.0.0 и маска /20. Маска преобразуется в маску DDN 255.255.240.0. При использовании сети класса B имеется 16 битов сети и только 4 бита подсети, находящихся в третьем октете. Таким образом, это случай с менее чем 8 битами подсети и третьим интересующим октетом.

Для начала запишите нулевую подсеть, а затем начинайте добавлять магическое число в интересующий октет. Нулевая подсеть равна идентификатору сети (в данном случае 172.30.0.0). Магическое число, вычисленное как 256 – 240 = 16, следует добавить к предыдущему интересующему октету идентификатора подсети. Результаты см. в табл. 19.8.

Таблица 19.8. Перечень всех подсетей: 172.30.0.0/2

Октет	1	2	3	4
Маска	255	255	240	0
Магическое число			16	
Номер сети/нулевая подсеть	172	30	0	0
Первая не нулевая подсеть	172	30	16	0
Следующая подсеть	172	30	32	0
Следующая подсеть	172	30	48	0
Следующая подсеть	172	30	64	0
Следующая подсеть	172	30	Пропуск...	0
Следующая подсеть	172	30	224	0
Широковещательная подсеть	172	30	240	0
Вне диапазона — используется процессом	172	30	256	0

Ответ на практическое задание 3

В третьем задании даны сеть 10.0.0.0 и маска /17. Маска преобразуется в маску DDN 255.255.128.0. При использовании сети класса A имеется 8 битов сети и 9 битов подсети. Интересующим является октет 3, только с 1 битом подсети в этом октете, а второй октет является просто левым октетом, с 8 битами подсети.

В данном случае начните с поиска первого блока подсетей. Магическое число 256 – 128 = 128. Первая подсеть (нулевая) равна идентификатору сети. Таким образом, первый блок идентификаторов подсети выглядит так:

10.0.0.0
10.0.128.0

Затем создайте блок подсетей для всех 256 возможных значений в просто левом октете, или октете 2 в данном случае. Следующий список демонстрирует первые три блока идентификаторов подсети плюс последний блок идентификаторов подсети, а не перечень на целую страницу:

10.0.0.0 (нулевая подсеть)
10.0.128.0
10.1.0.0
10.1.128.0
10.2.0.0
10.2.128.0
...
10.255.0.0
10.255.128.0 (широковещательная подсеть)

Ответы на контрольные вопросы:

1 А. **2** Б. **3** Б. **4** Г. **5** В и Г. **6** А. **7** Г. **8** Г.

ГЛАВА 20

Маски подсети переменной длины

В IPv4-адресации и создании подсетей используется много взаимосвязанных терминов, математических подходов и концепций. При изучении этих концепций не стоит их усложнять. Поэтому в данной книге до сих пор обсуждались, по возможности, наиболее простые случаи, когда используется только одна маска для всей сети класса A, B или C.

В этой главе данное ограничение снято, она представляет *маски подсети переменной длины* (Variable-Length Subnet Mask — VLSM). Маски VLSM позволяют использовать в проекте подсети несколько масок для той же классовой сети. У масок VLSM есть и преимущества, и недостатки, но основная сложность при их изучении в том, что проектирование подсетей с использованием масок VLSM требует больше математических вычислений, а также решения некоторых дополнительных задач. Эти концепции, задачи и математические подходы рассматриваются в данной главе.

В этой главе рассматриваются следующие экзаменационные темы

IP-адресация (IPv4/IPv6)

Выбор подходящей схемы IPv4-адресации (использующей VLSM и суммирование), удовлетворяющей требования адресации в среде LAN/WAN.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Маски VLSM, концепции и конфигурация

Маски VLSM применяются в тех случаях, когда в различных подсетях отдельной сети класса A, B или C возникает необходимость использования нескольких разных масок. На рис. 20.1 приведен пример использования маски VLSM в сети класса A с номером 10.0.0.0.

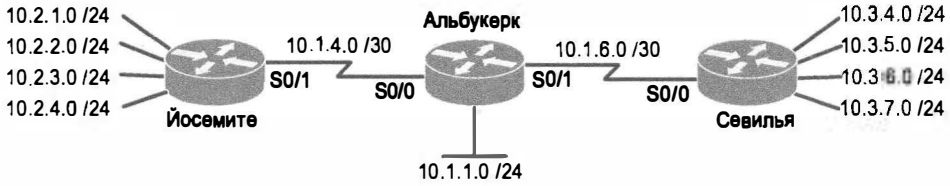


Рис. 20.1. Маска VLSM в сети с номером 10.0.0.0: маски /24 и /30

На рис. 20.1 показан типичный вариант использования префикса /30 (маска 255.255.255.252) в двухточечных последовательных каналах, в подсетях локальной сети, где используется другая маска (в данном случае /24 (255.255.255.0)). Все подсети относятся к сети класса A с номером 10.0.0.0, и в них используются две маски, поэтому ситуация соответствует определению понятия маски VLSM.

Следует отметить такую распространенную ошибку: многие специалисты считают, что маска VLSM означает "использование нескольких масок в одной сети", а не "использование нескольких масок *в отдельной классовой сети*". Например, если в схеме одной объединенной сети во всех подсетях сети 10.0.0.0 применяется маска 255.255.240.0, а во всех подсетях сети 11.0.0.0 — маска 255.255.255.0, то фактически используются две различные маски. Однако в сети класса A 10.0.0.0 используется только одна маска и в сети класса A 11.0.0.0 используется только одна маска. В этом случае маски VLSM не используются.

Маски VLSM обеспечивают много преимуществ для реальных сетей, главным образом в связи с тем, как резервируется и используется пространство IP-адресов. Поскольку маска определяет размер подсети (количество адресов хостов в подсети), маски VLSM позволяют лучше приспосабливаться к потребностям в адресах согласно размерам подсетей. Например, для подсетей, которые нуждаются в меньшем количестве адресов, инженер использует маску с меньшим количеством битов хоста, таким образом, в подсети оказывается меньше IP-адресов хостов. Эта гибкость сокращает количество IP-адресов, потраченных впустую в каждой подсети. При трате меньшего количества адресов остается больше пространства для создания больших подсетей.

Маски VLSM могут быть полезны и для открытых, и для частных IP-адресов, но в открытых сетях их преимущества более существенны. Экономя адреса в открытых сетях, инженеры избегают необходимости получать другой зарегистрированный номер сети IP у соответствующих организаций. В частных сетях, как определено в документе RFC, 1918, исчерпание адресов не является большой проблемой, поскольку по желанию всегда можно получить другую закрытую сеть.

Бесклассовые и классовые протоколы маршрутизации

Прежде чем можно будет развернуть проект VLSM, созданный на бумаге, необходимо сначала задействовать протокол маршрутизации, поддерживающий маски VLSM. Для поддержки масок VLSM протокол маршрутизации должен анонсировать маску наряду с каждой подсетью. Без информации о маске получающий обновления маршрутизатор будет введен в заблуждение.

Например, если маршрутизатор изучил маршрут для сети 10.1.8.0, то что он означает без информации о маске? Это подсеть 10.1.8.0/24?, 10.1.8.0/23? или 10.1.8.0/30? Десятично-точечное число 10.1.8.0 может означать любой номер подсети при допустимой маске, а поскольку с VLSM может использоваться несколько масок, у маршрутизатора нет никакого способа выяснить, какой конкретно. Чтобы фактически поддерживать маски VLSM, протокол маршрутизации должен анонсировать маску наряду с каждой подсетью, чтобы получающий маршрутизатор знал точно, какая подсеть анонсируется.

По определению *бесклассовые протоколы маршрутизации* (classless routing protocol) анонсируют маску в каждом анонсе маршрута, а *классовые протоколы маршрутизации* (classful routing protocol) — нет. Бесклассовые протоколы маршрутизации (табл. 20.1) являются более новыми и передовыми. Они не только обеспечивают расширенную поддержку масок VLSM, но и делают возможным суммирование маршрутов вручную (подробнее об этом — в главе 21).

Таблица 20.1. Бесклассовые и классовые протоколы маршрутизации IP внутреннего шлюза с учетом поддержки масок VLSM и суммирования

Ключевая тема

Протокол маршрутизации	Является бесклассовым	Передает маску в обновлениях	Поддерживает маски VLSM	Поддерживает суммирование маршрутов вручную
RIP-1	Нет	Нет	Нет	Нет
IGRP	Нет	Нет	Нет	Нет
RIP-2	Да	Да	Да	Да
EIGRP	Да	Да	Да	Да
OSPF	Да	Да	Да	Да

Кроме самих масок VLSM, протоколы маршрутизации не требуют никакой настройки для поддержки масок VLSM или чтобы быть бесклассовыми. Нет никакой команды, позволяющей запретить или разрешить бесклассовым протоколам маршрутизации включать маску в каждый маршрут. Единственный конфигурационный выбор, который можно сделать, подразумевает использование бесклассового протокола маршрутизации, которым, как упоминалось при обсуждении протокола маршрутизации, может быть протокол EIGRP или OSPF.

Настройка и проверка масок VLSM

В маршрутизаторах маски VLSM как средство настройки конфигурации и не запрещается, и не разрешается. С точки зрения настройки конфигурации маска VLSM представляет собой просто побочный эффект применения подкоманды интерфейса `ip address`. В маршрутизаторах настройка конфигурации маски VLSM выполняется на основании наличия IP-адресов в той же классовой сети, но с разными масками.

Пример 20.1 демонстрирует простой случай с двумя интерфейсами маршрутизатора Yosemite (см. рис. 20.1), а также назначение IP-адресов двум интерфейсам, один с маской /24 и один с маской /30, оба с IP-адресами в сети класса A 10.0.0.0.

Пример 20.1. Настройка двух интерфейсов на маршрутизаторе Yosemite с учетом масок VLSM

```
Yosemite#configure terminal
Yosemite(config)#interface Fa0/0
Yosemite(config-if)#ip address 10.2.1.1 255.255.255.0
Yosemite(config-if)#interface S0/1
Yosemite(config-if)#ip address 10.1.4.1 255.255.255.252
```

Использование масок VLSM можно также обнаружить при подробном рассмотрении вывода команды `show ip route`. Эта команда выводит маршруты сгруппированными по классовым сетям, чтобы продемонстрировать все подсети каждой сети класса A, B или C. Проверить количество разных масок, если таковые вообще имеются, можно только внизу вывода. Пример 20.2 демонстрирует содержимое таблицы маршрутизации маршрутизатора Albuquerque, показанного на рис. 20.1. Как уже упоминалось, выделенные строки примера демонстрируют использование маршрутизатором Albuquerque масок /24 и /30 в сети 10.0.0.0.

Пример 20.2. Таблица маршрутизации маршрутизатора Albuquerque с маской VLSM

```
Albuquerque# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
       level-2, ia - IS-IS inter area, * - candidate default,
       U - per-user static route, o - ODR
       P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set


Gateway of last resort is not set

        10.0.0.0/8 is variably subnetted, 14 subnets, 3 masks
D         10.2.1.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D         10.2.2.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D         10.2.3.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D         10.2.4.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D         10.3.4.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
D         10.3.5.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
D         10.3.6.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
D         10.3.7.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
C         10.1.1.0/24 is directly connected, FastEthernet0/0
L         10.1.1.1/32 is directly connected, FastEthernet0/0
C         10.1.6.0/30 is directly connected, Serial0/1
L         10.1.6.1/32 is directly connected, Serial0/1
C         10.1.4.0/30 is directly connected, Serial0/0
L         10.1.4.1/32 is directly connected, Serial0/0
```

ВНИМАНИЕ!

Чтобы понять, использует ли проект маски VLSM, игнорируйте локальные (L) маршруты /32, которые маршрутизатор автоматически создает для IP-адресов собственных интерфейсов.

На этом обсуждение масок VLSM самих по себе завершается. Данная глава посвящена маскам VLSM, но потребовалось всего три-четыре страницы, чтобы полностью описать их. Почему же для них понадобилась целая глава? Дело в том, что для работы с масками VLSM, поиска связанных с ними проблем, добавления подсетей к существующим проектам и разработке их применения, другими словами, применения масок VLSM в реальных условиях, необходимы навыки и практический опыт. Ответы на экзаменационные вопросы также потребуют навыков и практики. Остальная часть этой главы посвящена приобретению навыка применения масок VLSM, а также некоторым практическим задачам в следующих ключевых областях:

- Поиск перекрывающихся подсетей при использовании масок VLSM;
- Добавление новой подсети к существующему проекту VLSM.

Поиск перекрывающихся подсетей при использовании масок VLSM

Независимо от того, использует ли проект маски VLSM, адресные диапазоны подсетей, используемых в любом проекте объединенной сети IP, не должны перекрываться. Когда адреса различных подсетей перекрываются, перекрываются и записи таблиц маршрутизации маршрутизаторов. В результате хостам в разных местах могут быть присвоены одинаковые IP-адреса. В таких случаях маршрутизаторы, безусловно, не могут правильно перенаправить пакеты. Короче говоря, проект, в котором подсети перекрываются, никак не может считаться правильным и не должен использоваться.

ВНИМАНИЕ!

Хоть я и не видел применения этого термина в других местах книги, только для полноты описания упомяну противоположность масок VLSM — *маску подсети постоянной длины* (Static Length Subnet Mask — SLSM). Другими словами, единую маску, используемую повсюду в классовой сети.

Перекрытие адресов проще заметить при использовании масок SLSM, а не масок VLSM. У перекрывающихся подсетей при использовании масок SLSM идентичные идентификаторы подсети, поэтому для поиска перекрытий достаточно просмотреть идентификаторы подсети. У перекрывающихся подсетей при использовании масок VLSM идентификаторы подсети могут быть разными. Для поиска такого перекрытия придется просмотреть весь диапазон адресов в каждой подсети, от идентификатора подсети до ее широковещательного адреса, и сравнить диапазон с диапазонами других подсетей в проекте.

Пример поиска перекрытий VLSM

Предположим, например, что на экзамене CCENT встретился практический вопрос, представленный на рис. 20.2. Здесь показана отдельная сеть класса B (172.16.0.0) с масками VLSM, поскольку она использует три разные маски: /23, /24 и /30.

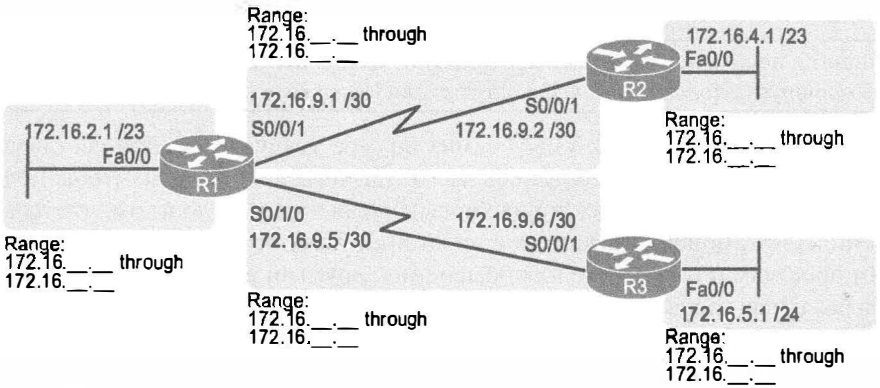


Рис. 20.2. Проект сети с маской VLSM, в которой могут возникать перекрытия

Теперь предположим, что в экзаменационном вопросе приведен рисунок, аналогичный рис. 20.2, и прямо или косвенно предъявлено требование определить, существуют ли перекрывающиеся подсети. В вопросе такого типа может быть просто указано, что в некоторых хостах не может быть выполнена эхо-проверка друг друга, а сама основная причина, обусловленная наличием перекрытий между некоторыми подсетями, может даже не упоминаться. Для поиска ответов на подобные вопросы можно применить следующую последовательность действий, которая, тем не менее, может оказаться трудоемкой.

Ключевая тема

Этапы анализа существующего проекта на предмет обнаружения перекрытий

Этап 1 Вычислить идентификаторы и широковещательные адреса для каждой подсети. (Это позволит определить диапазоны адресов во всех подсетях.)

Этап 2 Записать идентификаторы подсети в числовом виде (наряду с их широковещательными адресами)

Этап 3 Просмотреть список сверху вниз, сравнивая каждую пару соседних записей и выясняя, нет ли перекрытия диапазонов их адресов

В табл. 20.2 приведен пример результата первых двух этапов для задачи на рис. 20.2. Это список идентификаторов и широковещательных адресов подсетей в числовом виде, отсортированный по идентификатору подсети.

Таблица 20.2. Идентификаторы подсети и диапазоны адресов для примера, показанного на рис. 20.2

Местонахождение подсети	Номер подсети	Широковещательный адрес
Локальная сеть маршрутизатора R1	172.16.2.0	172.16.3.255
Локальная сеть маршрутизатора R2	172.16.4.0	172.16.5.255
Локальная сеть маршрутизатора R3	172.16.5.0	172.16.5.255
Последовательный канал между маршрутизаторами R1 и R2	172.16.9.0	172.16.9.3
Последовательный канал между маршрутизаторами R1 и R3	172.16.9.4	172.16.9.7

Этап 3 подразумевает довольно простое действие по сравнению с диапазонами адресов и определения того, не происходят ли где-либо перекрытия. Можно просто просмотреть список в целом, но если он уже создан, то можно также систематически исследовать его, просматривая каждую смежную пару.

Сначала обратите внимание в табл. 20.1 лишь на столбец номера подсети. В данном случае ни один из номеров подсетей не совпадает, но две выделенные записи действительно перекрываются.

Затем внимательно присмотритесь к локальным подсетям маршрутизаторов R2 и R3. Все адреса подсети 172.16.5.0/24 являются частью подсети 172.16.4.0/23. В данном случае из-за перекрытия проект недопустим, а одна из этих двух подсетей должна быть изменена.

Кстати, если две соседние записи списка перекрываются, сравните их также с остальными записями. Эти две подсети, уже отмеченные как перекрывающиеся, вполне могут перекрываться со следующими подсетями в списке. Рассмотрим, например, случай, когда в списке исследуемых на предмет перекрытия VLSM подсетей есть три записи:

10.1.0.0/16 (идентификатор подсети 10.1.0.0, широковещательный адрес 10.1.255.255);

10.1.200.0/24 (идентификатор подсети 10.1.200.0, широковещательный адрес 10.1.200.255);

10.1.250.0/24 (идентификатор подсети 10.1.250.0, широковещательный адрес 10.1.250.255).

Если сравнить записи 1 и 2, вполне очевидно наличие перекрытия, так как все адреса в подсети 10.1.200.0/24 находятся в подсети 10.1.0.0/16. Если затем сравнить только записи 2 и 3, то перекрытия нет. Однако на самом деле записи 1 и 3 перекрываются. Что же это означает? Каждый раз, найдя перекрытие, сравнивайте все перекрывающиеся подсети со всеми следующими в списке подсетей, пока не найдете ту, где перекрытие заканчивается.

Практические задания по поиску перекрытия VLSM

Ниже приведены типичные практические задания на применение IP-адресации и создание подсетей. Табл. 20.3 содержит три практических задания. Здесь представлено по пять IP-адресов в каждом столбце. Используя описанный в предыдущем разделе трехэтапный процесс, найдите все перекрытия VLSM. Ответы приведены в конце главы, в разделе "Ответы на практические задания по поиску перекрытия VLSM".

Таблица 20.3. Практические задания по поиску перекрытия VLSM

Задача 1	Задача 2	Задача 3
10.1.34.9/22	172.16.126.151/22	192.168.1.253/30
10.1.29.101/23	172.16.122.57/27	192.168.1.113/28
10.1.23.254/22	172.16.122.33/30	192.168.1.245/29
10.1.17.1/21	172.16.122.1/30	192.168.1.125/30
10.1.1.1/20	172.16.128.151/20	192.168.1.122/30

Добавление новой подсети к существующему проекту VLSM

Задание, описанное в этом разделе, нередко встречается в реальных сетях: выбор новых подсетей, добавляемых в существующий проект. В реальной жизни можно использовать инструментальные средства, помогающие выбрать новую подсеть без перекрытия. Однако и в реальной жизни, и на экзамене CCNA необходимо быть готовым выполнить в уме вычисления по выбору подсети, которая имеет правильные IP-адреса для необходимого количества хостов без перекрытия подсетей. Другими словами, необходимо выбрать новую подсеть и не сделать ошибку!

В качестве примера рассмотрим объединенную сеть, представленную на рис. 20.2, в состав которой входит классовая сеть 172.16.0.0. На экзамене могут встретиться вопросы, которые указывают, что к проекту должна быть добавлена новая подсеть, допустим, с длиной префикса /23. В вопросе может быть также указано: "Выберите наименьший номер подсети, который может использоваться для новой подсети". Другими словами, если бы подходили две сети, 172.16.4.0 и 172.16.6.0, то выбрать следует сеть 172.16.4.0.

Таким образом, заданий здесь несколько: найти все применимые идентификаторы подсетей, исключить те, которые привели бы к перекрытию, а затем, если этого требует вопрос, выяснить самый низкий или самый высокий (в цифровой форме) идентификатор подсети. Этапы выполнения этого задания приведены ниже.

Ключевая тема

Этапы процесса добавления новой подсети к существующему проекту VLSM

Этап 1 Выберите маску подсети (длину префикса) для новой подсети на основании требований проекта (если она не указана в вопросе)

Этап 2 Вычислите все возможные номера подсетей классовой сети, используя найденную на этапе 1 маску, наряду с широковещательными адресами подсетей

Этап 3 Создайте список существующих идентификаторов и соответствующих широковещательных адресов подсети

Этап 4 Исключите новые перекрывающиеся подсети, сравнив списки из двух предыдущих этапов

Этап 5 Выберите новый идентификатор подсети из оставшихся подсетей, выявленных на этапе 4, с учетом того, требует ли вопрос представить самый низкий или самый высокий по числовому значению идентификатор подсети

Пример добавления новой подсети VLSM

Например, на рис. 20.3 показана существующая объединенная сеть, использующая маски VLSM. (Рисунок использует те же IP-адреса, что и на рис. 20.2, но с IP-адресом LAN маршрутизатора R3, измененным так, чтобы исправить перекрытие масок VLSM, представленное на рис. 20.2.) В данном случае необходимо добавить новую подсеть, содержащую 300 хостов. Предположим, в вопросе сказано, что следует использовать наименьшую подсеть (с наименьшим количеством хостов), удовлетворяющую этому требованию. Используя описанный ранее математический механизм и логику, выбираем маску /23, которая дает 9 битов хоста для $2^9 - 2 = 510$ хостов в подсети.

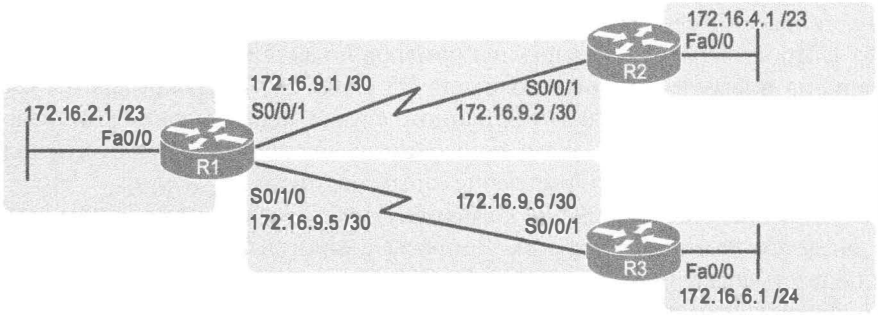


Рис. 20.3. Объединенная сеть 172.16.0.0, к который необходимо добавить подсеть с маской /23

ВНИМАНИЕ!

Если логика и математический механизм, упомянутые в предыдущем разделе, вам незнакомы, то имеет смысл обратиться к главе 19.

На настоящий момент достаточно следовать этапам, перечисленным до рис. 20.3. На этапе 1 уже была найдена маска (/23). Для этапа 2 необходимо записать все номера подсетей классовой сети 172.16.0.0 и их широковещательные адреса с учетом маски /23. Использовать все эти подсети не придется, но необходим список для сравнения с существующими подсетями. В табл. 20.3 приведены результаты для первых пяти возможных подсетей /23.

Таблица 20.4. Первые пять возможных подсетей /23

Подсеть	Номер подсети	Широковещательный адрес
Первая (нулевая)	172.16.0.0	172.16.1.255
Вторая	172.16.2.0	172.16.3.255
Третья	172.16.4.0	172.16.5.255
Четвертая	172.16.6.0	172.16.7.255
Пятая	172.16.8.0	172.16.9.255

Далее, на этапе 3, перечислите существующие номера подсетей и их широковещательные адреса, как отмечено на рис. 20.3. Чтобы, взяв IP-адрес и маску, получить идентификатор подсети и ее широковещательный адрес, достаточно обычной математики. Полученная информация, включая расположение, номера подсети и широковещательные адреса, приведена в табл. 20.5.

Таблица 20.5. Существующие идентификаторы подсети и их широковещательные адреса согласно рис. 20.3

Подсеть	Номер подсети	Широковещательный адрес
Локальная сеть маршрутизатора R1	172.16.2.0	172.16.3.255
Локальная сеть маршрутизатора R2	172.16.4.0	172.16.5.255
Локальная сеть маршрутизатора R3	172.16.6.0	172.16.6.255
Последовательный канал между R1 и R2	172.16.9.0	172.16.9.3
Последовательный канал между R1 и R3	172.16.9.4	172.16.9.7

На настоящий момент есть вся информация, необходимая для поиска перекрытий на этапе 4. Просто сравните диапазоны номеров подсетей в двух предыдущих таблицах. Какие из новых возможных подсетей /23 (табл. 20.4) перекрываются с существующими подсетями (табл. 20.5)? В данном случае перекрываются вторая, третья и пятая подсети в табл. 20.4, поэтому исключим их из списка кандидатов на использование. (В табл. 20.4 эти подсети выделены серым цветом.)

Этап 5 имеет больше отношение к экзамену, чем к реальным проектам сети, но он все же является отдельным этапом. Вопросы с многовариантным выбором ответа иногда подразумевают один ответ и требуют указать самую низкую или самую высокую (в цифровой форме) подсеть. В данном конкретном задании нужен самый низкий номер подсети (в цифровой форме), которым в данном случае является 172.16.0.0/23.

ВНИМАНИЕ!

Ответ 172.16.0.0/23 может оказаться нулевой подсетью. Что касается экзамена, то применения нулевой подсети следует избегать, если, во-первых, из формулировки экзаменационного вопроса следует, что используется классовый протокол маршрутизации, или, во-вторых, что настройка конфигурации маршрутизаторов выполнена с помощью глобальной команды конфигурации `no ip subnet-zero`. В противном случае предполагается, что может применяться нулевая подсеть.

Практические задания по добавлению новых подсетей VLSM

Руководство требует добавить в существующий проект еще одну подсеть. Существующий проект уже насчитывает пять подсетей:

10.0.0.0/24
10.0.1.0/25
10.0.2.0/26
10.0.3.0/27
10.0.6.0/28

Руководство не может решить, какую из пяти альтернативных масок подсети использовать для этой следующей новой подсети, добавляемой в объединенную сеть. Но руководство желает, чтобы вы попрактиковались в масках VLSM и вычислили идентификаторы подсети, применимые для каждой из тех четырех возможных масок. Новый идентификатор подсети должен быть частью сети класса A 10.0.0.0, новая подсеть не должна перекрывать ни одну из пяти первоначальных подсетей, а кроме того, новый идентификатор подсети должен быть наименьшим возможным (без нарушения других правил). На основании каждой из следующих возможных масок выберите по одному подходящему идентификатору подсети:

1. /24
2. /23
3. /22
4. /25
5. /26

Ответы приведены в конце главы, в разделе "Ответы на практические задания по поиску перекрытия VLSM".

Обзор

Резюме

- Маски подсети переменной длины (VLSM) позволяют использовать в проекте подсети несколько масок для той же классовой сети.
- Маски VLSM обеспечивают много преимуществ для реальных сетей, главным образом в связи с тем, как резервируется и используется пространство IP-адресов. Поскольку маска определяет размер подсети (количество адресов хостов в подсети), маски VLSM позволяют лучше приспосабливаться к потребностям в адресах согласно размерам подсетей.
- Маски VLSM могут быть полезны и для открытых, и для частных IP-адресов, но в открытых сетях их преимущества более существенны. Экономя адреса в открытых сетях, инженеры избегают необходимости получать другой зарегистрированный номер сети IP у соответствующих организаций.
- Для поддержки масок VLSM протокол маршрутизации должен анонсировать маску наряду с каждой подсетью. Без информации о маске получающий обновления маршрутизатор будет введен в заблуждение.
- По определению бесклассовые протоколы маршрутизации анонсируют маску в каждом анонсе маршрута, а классовые протоколы маршрутизации — нет.
- В маршрутизаторах маски VLSM как средство настройки конфигурации и не запрещается, и не разрешается. С точки зрения настройки конфигурации маска VLSM представляет собой просто побочный эффект применения подкоманды интерфейса `ip address`. В маршрутизаторах настройка конфигурации маски VLSM выполняется на основании наличия IP-адресов в той же классовой сети, но с разными масками.
- Использование масок VLSM подразумевает пять основных этапов. Необходимо найти все применимые идентификаторы подсетей, исключить те, которые привели бы к перекрытию, а затем, если этого требует вопрос, выяснить самый низкий или самый высокий (в цифровой форме) идентификатор подсети. Этапы решения этой задачи приведены ниже.

  Этап 1 Выберите маску подсети (длину префикса) для новой подсети на основании требований проекта (если она не указана в вопросе)

  Этап 2 Вычислите все возможные номера подсетей классовой сети, используя найденную на этапе 1 маску, наряду с широковещательными адресами подсетей

  Этап 3 Создайте список существующих идентификаторов и соответствующих широковещательных адресов подсети

  Этап 4 Исключите новые перекрывающиеся подсети, сравнив списки из двух предыдущих этапов

  Этап 5 Выберите новый идентификатор подсети из оставшихся подсетей, выявленных на этапе 4, с учетом того, требует ли вопрос представить самый низкий или самый высокий по числовому значению идентификатор подсети

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Какие из указанных протоколов маршрутизации поддерживают маски VLSM? (Выберите три ответа.)

   А) RIP-1.
   Б) RIP-2.
   В) EIGRP.
   Г) OSPF.

2. Как расшифровывается аббревиатура VLSM?

   А) Variable length subnet mask (маска подсети переменной длины).
   Б) Very long subnet mask (очень длинная маска подсети).
   В) Vociferous longitudinal subnet mask (многословная продольная маска подсети).
   Г) Vector-length subnet mask (маска подсети векторной длины).
   Д) Vector loop subnet mask (векторная замкнутая маска подсети).

3. В маршрутизаторе R1 настройка конфигурации интерфейса Fa0/0 выполнена с помощью команды `ip address 10.5.48.1 255.255.240.0`. Какая из следующих подсетей после настройки ее конфигурации в другом интерфейсе маршрутизатора R1 не будет рассматриваться как подсеть с перекрывающейся маской VLSM?

   А) 10.5.0.0 255.255.240.0.
   Б) 10.4.0.0 255.254.0.0.
   В) 10.5.32.0 255.255.224.0.
   Г) 10.5.0.0 255.255.128.0.

4. Маршрутизатор R4 имеет маршрут для подключенной сети 172.16.8.0/22. В каком из следующих ответов указана подсеть, перекрывающаяся с этой подсетью?

   А) 172.16.0.0/21
   Б) 172.16.6.0/23
   В) 172.16.16.0/20
   Г) 172.16.11.0/25

5. Проект уже включает подсети 192.168.1.0/26, 192.168.1.128/30 и 192.168.1.160/29. Какая из следующих подсетей имеет самым низкий (в цифровой форме) идентификатор подсети, который мог быть добавлен в проект, если бы было нужно добавить подсеть, использующую маску /28?

   А) 192.168.1.144/28
   Б) 192.168.1.112/28
   В) 192.168.1.64/28
   Г) 192.168.1.80/28
   Д) 192.168.1.96/28

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 20.6.

Таблица 20.6. Ключевые темы главы 20

Элемент	Описание	Страница
Табл. 20.1	Бесклассовые и классовые протоколы маршрутизации IP внутреннего шлюза с учетом поддержки масок VLSM и суммирования	605
Список	Этапы анализа существующего проекта на предмет обнаружения перекрытий	608
Список	Этапы процесса добавления новой подсети к существующему проекту VLSM	610

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

Классовый протокол маршрутизации (classful routing protocol), бесклассовый протокол маршрутизации (classless routing protocol), перекрывающаяся подсеть (overlapping subnets), маски подсети переменной длины (Variable-Length Subnet Masks — VLSM)

Практические задания в приложении 3

Дополнительные практические задания и ответы на них приведены в приложении 3. Это приложение можете найти на веб-странице книги в формате PDF.

Практика

Ответы на практические задания по поиску перекрытия VLSM

В этом разделе даны ответы на три практических задания раздела "Практические задания по поиску перекрытия VLSM", перечисленных в табл. 20.3. Обратите внимание: подсети в таблицах с подробностями ответов уже переупорядочены согласно процессу.

В задании 1 идентификаторы второй и третий подсетей, указанных в табл. 20.7, перекрываются. Диапазон второй подсети полностью включает диапазон адресов третьей подсети.

Таблица 20.7. Ответы на задание 1 по поиску перекрытия VLSM (перекрытия выделены)

Подсеть	Исходный адрес и маска	Идентификатор подсети	Широковещательный адрес
1	10.1.1.1/20	10.1.0.0	10.1.15.255
2	10.1.17.1/21	10.1.16.0	10.1.23.255
3	10.1.23.254/22	10.1.20.0	10.1.23.255
4	10.1.29.101/23	10.1.28.0	10.1.29.255
5	10.1.34.9/22	10.1.32.0	10.1.35.255

В задании 2 идентификаторы второй и третьей подсетей (см. табл. 20.8) также перекрываются. Диапазон адресов второй подсети полностью включает диапазон адресов третьей подсети. Кроме того, идентификаторы второй и третьей подсетей совпадают, таким образом, это вполне очевидное перекрытие.

Таблица 20.8. Ответы на задание 2 по поиску перекрытия VLSM (перекрытия выделены)

Подсеть	Исходный адрес и маска	Идентификатор подсети	Широковещательный адрес
1	172.16.122.1/30	172.16.122.0	172.16.122.3
2	172.16.122.57/27	172.16.122.32	172.16.122.63
3	172.16.122.33/30	172.16.122.32	172.16.122.35
4	172.16.126.151/22	172.16.124.0	172.16.127.255
5	172.16.128.151/20	172.16.128.0	172.16.143.255

В задании 3 перекрываются три подсети. Диапазон подсети 1 полностью включает диапазон адресов второй и третьей подсетей. Обратите внимание на то, что вторая и третья подсети не перекрываются, таким образом, как было упомянуто в этой книге, при поиске всех перекрытий после обнаружения перекрытия первых двух подсетей следует сравнить следующую запись в таблице (3) с обеими из двух уже известных записей с перекрытием (1 и 2).

Таблица 20.9. Ответы на задание 3 по поиску перекрытия VLSM (перекрытия выделены)

Подсеть	Исходный адрес и маска	Идентификатор подсети	Широковещательный адрес
1	192.168.1.113/28	192.168.1.112	192.168.1.127
2	192.168.1.122/30	192.168.1.120	192.168.1.123
3	192.168.1.125/30	192.168.1.124	192.168.1.127
4	192.168.1.245/29	192.168.1.240	192.168.1.247
5	192.168.1.253/30	192.168.1.252	192.168.1.255

Ответы на практические задания по добавлению новых подсетей VLSM

В этом разделе приведены ответы на четыре практических задания раздела "Практические задания по добавлению новых подсетей VLSM".

Во всех пяти заданиях этого раздела использован одинаковый набор из пяти существовавших ранее подсетей. Список идентификаторов подсетей и их широковещательных адресов, определяющих верхнюю и нижнюю границы диапазона адресов каждой подсети, приведен в табл. 20.10.

Таблица 20.10. Существующие подсети для заданий данной главы по добавлению подсетей VLSM

Подсеть	Номер подсети	Широковещательный адрес
1	10.0.0.0/24	10.0.0.255
2	10.0.1.0/25	10.0.1.127
3	10.0.2.0/26	10.0.2.63
4	10.0.3.0/27	10.0.3.31

Далее следуют объяснения согласно процессу, описанному ранее, в разделе "Добавление новой подсети к существующему проекту VLSM", за исключением того, что объяснения игнорируют этап 3, поскольку его результаты в каждом случае уже перечислены в табл. 20.10.

Задание 1

Этап 1 Формулировка задания требует использовать маску /24

Этап 2 Возможными подсетями были бы: 10.0.0.0, 10.0.1.0, 10.0.2.0, 10.0.3.0, 10.0.4.0, 10.0.5.0 и так далее, на 1 больше в третьем октете

Этап 3 Все четыре из первых новых возможных подсетей (10.0.0.0/24, 10.0.1.0/24, 10.0.2.0/24 и 10.0.3.0/24) перекрываются с существующими подсетями (см. табл. 20.10). Подсеть 10.0.6.0/24 также перекрывается

Этап 4 Самый низкий новый номер подсети (в цифровой форме), который не перекрывается с существующими подсетями, — 10.0.4.0/24

Задание 2

Этап 1 Формулировка задания требует использовать маску /23

Этап 2 Возможными подсетями были бы: 10.0.0.0, 10.0.2.0, 10.0.4.0, 10.0.6.0, 10.0.8.0 и так далее, на 2 больше в третьем октете

Этап 3 Все три первых из четырех новых возможных подсетей (10.0.0.0/23, 10.0.2.0/23 и 10.0.6.0/23) перекрываются с существующими подсетями

Этап 4 Самый низкий новый номер подсети (в цифровой форме), который не перекрывается с существующими подсетями, — 10.0.4.0/23

Задание 3

Этап 1 Формулировка задания требует использовать маску /22

Этап 2 Возможными подсетями были бы: 10.0.0.0, 10.0.4.0, 10.0.8.0, 10.0.12.0 и так далее, на 4 больше в третьем октете

Этап 3 Первые две новых возможных подсети (10.0.0.0/22, 10.0.4.0/22) перекрываются с существующими подсетями

Этап 4 Самый низкий новый номер подсети (в цифровой форме), который не перекрывается с существующими подсетями, — 10.0.8.0/22

Задание 4

Ответ на это задание требует больше подробностей, чем другие, поскольку маска /25 создает больше подсетей, которые могли бы перекрываться с существующими ранее подсетями. Вы уже знаете, что для этого задания на этапе 1 будет использоваться маска /25. В табл. 20.11 приведен результат этапа 2, содержащий первые 14 подсетей сети 10.0.0.0, с использованием маски /25. На этапе 4 в табл. 20.11 были выделены перекрывающиеся подсети. Чтобы завершить задачу на этапе 5, просмотрите последовательно таблицу и найдите первую неперекрывающуюся запись, 10.0.1.128/25.

Таблица 20.11. Первые 14 подсетей сети 10.0.0.0 при использовании маски /25

Подсеть	Номер подсети	Широковещательный адрес
1	10.0.0.0	10.0.0.127
2	10.0.0.128	10.0.0.255
3	10.0.1.0	10.0.1.127
4	10.0.1.128	10.0.1.255
5	10.0.2.0	10.0.2.127
6	10.0.2.128	10.0.2.255
7	10.0.3.0	10.0.3.127
8	10.0.3.128	10.0.3.255
9	10.0.4.0	10.0.4.127
10	10.0.4.128	10.0.4.255
11	10.0.5.0	10.0.5.127
12	10.0.5.128	10.0.5.255
13	10.0.6.0	10.0.6.127
14	10.0.6.128	10.0.6.255

Ответы на контрольные вопросы:

1 Б, В и Г. **2** А. **3** А. **4** Г. **5** В.

ГЛАВА 21

Суммирование маршрутов

Инструментальные средства суммирования маршрутов позволяют инженерам анонсировать один маршрут, заменяющий несколько других маршрутов новым, соответствующим тому же диапазону адресов. Это позволяет снизить нагрузку и сэкономить ресурсы процессора, полосу пропускания, а также не растрачивать память впустую.

В первой части данной главы рассматривается суммирование маршрутов вручную. Концепция суммирования маршрутов полагается на те же математические принципы, что и создание подсетей: она требует тщательного планирования подсетей, предусматривающего последующие попытки суммирования маршрутов. В первом разделе данной главы описаны концепции и математический механизм. Второй раздел посвящен систематическому способу поиска наилучших суммарных маршрутов, используемому при настройке суммарных маршрутов.

В этой главе рассматриваются следующие экзаменационные темы

IP-адресация (IPv4/IPv6)

Выбор подходящей схемы IPv4-адресации (использующей VLSM и суммирование), удовлетворяющей требования адресации в среде LAN/WAN.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Концепции суммирования маршрутов вручную

В маршрутизаторах небольших сетей таблицы маршрутизации могут содержать всего лишь несколько десятков маршрутов. Но у некоторых предприятий могут быть десятки тысяч подсетей, если не больше. Даже при существенном снижении количества маршрутов за счет суммирования, таблицы BGP маршрутизатора Интернета передали 450 тысяч меток, как было зарегистрировано в 2012 году.

По мере роста таблицы маршрутизации у маршрутизатора могут возникать проблемы. Сами таблицы занимают память маршрутизатора. Маршрутизация (перенаправление пакета) требует, чтобы маршрутизатор распознал маршрут в таблице маршрутизации, а поиск в более длинной таблице занимает и больше времени, и ресурсов процессора. Протоколам маршрутизации требуется больше усилий для обработки маршрутов и большая полоса пропускания для передачи анонсов маршрутов. При большой таблице маршрутизации это занимает больше времени, а поиск причин проблем затрудняется, поскольку инженерам, занимающимся отладкой сети, приходится просматривать больший объем информации.

Суммирование маршрутов позволяет сетевым инженерам преодолевать некоторые из этих проблем за счет замены многих маршрутов к меньшим подсетям одним маршрутом к тому, что выглядит как большая подсеть. Этот раздел знакомит с основами суммирования маршрутов и его влиянием на маршрутизаторы в сети IPv4. Кроме того, будет показано, что план создания подсетей должен загодя учитывать потребность в суммировании маршрутов, и описана проверка суммирования маршрутов при помощи команды `show ip route`.

ВНИМАНИЕ!

В этой главе под суммированием маршрутов подразумевается *суммирование маршрутов вручную* (manual route summarization), отличное от *автоматического суммирования* (autosummarization). Из них суммирование вручную — это фактический инструмент, используемый большинством сетевых инженеров, в то время как термин *автоматическое суммирование* относится к способу работы некоторых устаревших протоколов маршрутизации. Термин *вручную* означает, что инженер сам вводит одну или несколько команд, создающих суммарный маршрут. *Автоматическое суммирование*, создающее суммарные маршруты для решения проблем некоторых протоколов маршрутизации, обсуждается во втором томе книге.

Основы суммирования маршрутов

Представьте себе небольшой маршрутизатор с ограниченными возможностями процессора и скромной памятью, установленный в большой корпоративной сети, которая насчитывает более 10 000 подсетей. Небольшой маршрутизатор покорно изучает все маршруты, используя свой протокол маршрутизации, и добавляет их в свою таблицу маршрутизации. Это занимает его память; а протоколам маршрутизации из-за большого объема требуется больше работы. Кроме того, слишком длинная таблица маршрутизации означает более длительное время поиска соответствующего маршрута.

У большинства из этих 10 000 маршрутов одинаковые инструкции перенаправления: послать пакет на тот же конкретный интерфейс, в направлении ядра корпоративной сети. Разве не было бы прекрасно, если бы вместо нескольких тысяч маршрутов у этого маршрутизатора был бы только один маршрут, соответствующий всем этим инструкциям на перенаправление пакетов на тот же интерфейс? Именно это и делает суммирование маршрутов.

Суммирование маршрутов позволяет настроить протокол маршрутизации так, чтобы он анонсировал один маршрут вместо нескольких. Этот процесс создает новый суммарный маршрут, соответствующий некому диапазону адресов первоначальных маршрутов. Например, вместо анонсирования маршрутов для множества подсетей /24, таких как 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24 и т.д., маршрутизатор мог бы просто анонсировать маршрут для сети 172.16.0.0/16 и не анонсировать все меньшие подсети.

У суммирования маршрутов есть много преимуществ. Оно сокращает размер таблиц маршрутизации, тем не менее позволяя маршрутизатору перенаправлять пакеты ко всем получателям в сети. Более короткая таблица позволяет повысить производительность маршрутизации и сэкономить память на каждом маршрутизаторе. Суммирование улучшает также время конвергенции для протоколов маршрутизации, поскольку у них существенно снизится объем работы.

Суммирование маршрутов и план создания подсетей IPv4

Для лучшей работы план подсетей IPv4 следует создавать с учетом суммирования маршрутов. Суммирование объединяет несколько маршрутов в один, но чтобы оно сработало, первоначальные маршруты должны находиться в том же числовом диапазоне. Это может произойти случайно, но лучше спланировать заранее. На рис. 21.1 приведен пример типичной объединенной сети с двумя наборами из четырех подсетей, подлежащих суммированию (одни слева, другие справа). Обратите внимание, что план подсетей располагает подсети, начинающиеся на 10.2, слева, а начинающиеся на 10.3 — справа. Это существенно упрощает суммирование маршрутов. Чтобы увидеть почему, сосредоточимся на подсетях справа и проигнорируем пока подсети слева. Рисунок демонстрирует состояние сети перед суммированием маршрутов, изученных маршрутизатором R1 для подсетей справа.

Таблица маршрутизации R1

Подсеть	Выходной интерфейс	Следующий узел
10.3.4.0 /24	S0/1	10.1.6.3
10.3.5.0 /24	S0/1	10.1.6.3
10.3.6.0 /24	S0/1	10.1.6.3
10.3.7.0 /24	S0/1	10.1.6.3

Рис. 21.1. Небольшая объединенная сеть с хорошими кандидатами на суммирование маршрутов

Суммирование маршрутов вручную заставит маршрутизатор прекратить анонсировать набор маршрутов, анонсируя вместо них единый маршрут, содержащий набор всех адресов. Для этого маршрутизатор, создающий суммарный маршрут, следует настроить так, чтобы он знал анонсируемый номер подсети и маску. Протокол маршрутизации прекращает анонсировать старые маршруты (называемые *зависимыми маршрутами* (subordinate route)), анонсируя теперь только суммарный маршрут.

Рис. 21.2 продолжает пример, начатый на рис. 21.1, демонстрировавшем результат суммирования маршрута на маршрутизаторе R3. Этот суммарный маршрут заменяет маршруты для всех четырех подсетей справа. Только для упрощения математики суммарный маршрут использует подсеть 10.3.0.0/16. Обратите внимание, что подсеть 10.3.0.0/16 действительно включает все четыре первоначальные подсети, представленные на рис. 21.1 (а также другие адреса).

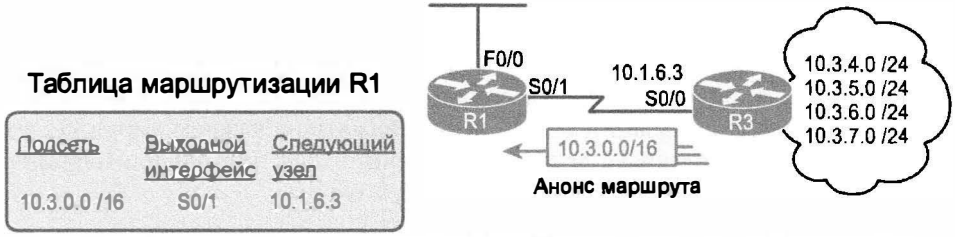


Рис. 21.2. Маршруты для четырех подсетей справа суммированы в один маршрут

При создании конфигурации суммарного маршрута на маршрутизаторе R3 маршрутизатор R1 (и другие маршрутизаторы далее по сети) также получает преимущество. Таблица маршрутизации маршрутизатора R1 уменьшается в размере, но что еще более важно, маршрутизатор R1 все еще может перенаправить пакеты тем же первоначальным четырем подсетям, на тот же интерфейс S0/1, тому же следующему транзитному маршрутизатору (10.1.6.3, которым является маршрутизатор R3).

Проверка маршрутов, полученных вручную

Суммирование маршрутов влияет на таблицы маршрутизации маршрутизаторов с различными результатами, в зависимости от того, изучил ли маршрутизатор суммарный маршрут или создал его. Пример 21.1 демонстрирует таблицы маршрутизации маршрутизатора R1 сначала до настройки суммарного маршрута на маршрутизаторе R3 (см. рис. 21.1), а затем после того, как маршрутизатор R3 добавил в конфигурацию суммарный маршрут (см. рис. 21.2). (Обратите внимание, что пример демонстрирует только маршруты, изученные протоколом маршрутизации, и не включает подключенные маршруты.)

Пример 21.1. Таблица маршрутизации маршрутизатора R1 до и после изучения суммарного маршрута

```
! Сначала до
R1# show ip route rip
! (Строки легенды пропущены для краткости)
     10.0.0.0/8 is variably subnetted, 14 subnets, 3 masks
R       10.2.1.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.2.2.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
```

```
R       10.2.3.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.2.4.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.3.4.0/24 [120/1] via 10.1.6.3, 00:00:12, Serial0/1
R       10.3.5.0/24 [120/2] via 10.1.6.3, 00:00:12, Serial0/1
R       10.3.6.0/24 [120/3] via 10.1.6.3, 00:00:12, Serial0/1
R       10.3.7.0/24 [120/4] via 10.1.6.3, 00:00:12, Serial0/1


! Теперь после
R1# show ip route
! (Строки легенды пропущены для краткости)

     10.0.0.0/8 is variably subnetted, 11 subnets, 4 masks
R       10.2.1.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.2.2.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.2.3.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.2.4.0/24 [120/1] via 10.1.4.2, 00:00:20, Serial0/0
R       10.3.0.0/16 [120/1] via 10.1.6.3, 00:00:04, Serial0/1
```

Сначала рассмотрим верхнюю половину вывода, демонстрирующую первоначальное состояние (на основании рис. 21.1). Команда `show ip route rip` вывела только изученные маршруты RIP, вывела статистику, утверждающую, что маршрутизатору R1 известно 14 подсетей, а затем перечислила восемь маршрутов, изученных по *протоколу маршрутной информации* (Routing Information Protocol — RIP). Другие шесть маршрутов — это три подключенных маршрута маршрутизатора R1 и три локальных маршрута для трех его интерфейсов. В частности, обратите внимание на четыре изученных маршрута RIP к подсетям, начинающимся на 10.3.

Затем, во второй части примера, представлено последующее состояние. Конечно, различие существенное — сначала было четыре индивидуальных подсети, начинающихся на 10.3, а после только суммарный маршрут для подсети 10.3.0.0/16 вместо них. Этот суммарный маршрут выглядит как любой другой маршрут с подсетью, маской, следующим транзитным маршрутизатором (10.1.6.3) и исходящим интерфейсом (Serial0/1). Кроме того, ничто в строке не указывает, что этот маршрут суммарный, он ничем не отличается от другой подсети, существующей где-нибудь в объединенной сети.

Настройка суммарных маршрутов не обсуждается ни далее в этой книге, ни в книге по ICND2. Сосредоточьтесь на суммировании маршрутов здесь, постарайтесь понять основные идеи, уяснить преимущества и подготовиться к работе с маршрутами, которые вполне могут оказаться суммарными. Для этого во второй половине данной главы будет рассмотрен математический механизм, связанный с суммарными маршрутами, а также выбор наилучшей подсети и маски для суммарного маршрута.

Выбор наилучших суммарных маршрутов

Суммирование маршрутов вручную работает лучше, когда сначала создается план подсетей, учитывающий суммирование. Например, в примерах на рис. 21.1 и 21.2 использовался хорошо продуманный план, где инженеры использовали подсети, начинающиеся только с 10.2 для маршрутизатора R2, и подсети, начинающиеся с10.3 для маршрутизатора R3.

При создании суммарного маршрута сетевой инженер вводит команду конфигурации, задающую подсеть и маску. Для этого годится множество комбинаций подсети и маски; однако не все из них будут наилучшим выбором. Слово *наилучшие* применительно к выбору суммарного маршрута означает, что он должен включать все указанные в вопросе подсети, но *с как можно меньшим количеством других адресов.* Таким образом, под наилучшим суммарным маршрутом в этой книге понимается следующее.

Ключевая тема

Критерии, определяющие суммарный маршрут как наилучший для данного набора подсетей

Суммарный маршрут с наименьшим диапазоном адресов, который включает все адреса всех подсетей, которые необходимо объединить в один суммарный маршрут.

В приведенном ранее примере суммирования подсети 10.3.4.0/24, 10.3.5.0/24, 10.3.6.0/24 и 10.3.7.0/24 совместно определяют диапазон адресов от 10.3.4.0 до 10.3.7.255. Суммарный маршрут (10.3.0.0/16) в примере 21.1 вполне работоспособен. Но он имеет множество IP-адресов, не относящихся к четырем первоначальным подсетям, поскольку включает диапазон от 10.3.0.0 до 10.3.255.255. Кроме того, диапазон суммарного маршрута 10.3.4.0/22 точно соответствует диапазону для этих четырех подсетей (10.3.4.0–10.3.7.255). По приведенному здесь определению маршруты 10.3.0.0/16 и 10.3.4.0/22 вполне работоспособны, но наилучшим суммарным маршрутом будет 10.3.40/22.

В следующем разделе будет показано, как получить набор допустимых суммарных маршрутов и выбрать наилучший.

Процесс поиска наилучшего суммарного маршрута

Для поиска наилучшего суммарного маршрута можно использовать метод проб и ошибок, метод обоснованных предположений, калькулятор подсетей или любой другой метод по вашему выбору. В целях подготовки к экзаменам CCENT и CCNA для поиска наилучшего суммарного маршрута имеет смысл использовать упрощенный десятичный процесс. Он подразумевает использование уже имеющихся навыков: поиск идентификатора, маски и широковещательного адреса подсети, как обсуждалось ранее в главе 14. Если эти математические вычисления получается выполнять уверенно, то процесс не должен вызывать проблем. (В противном случае повторите материал главы 14.)

Ниже приведены этапы поиска наилучшего суммарного маршрута с использованием десятичного математического подхода, а также несколькими последующими примерами.

Ключевая тема

Процесс поиска наилучшего маршрута при суммировании вручную

Этап 1 Перечислите все номера подлежащих суммированию (зависимых) подсетей в десятичном виде в порядке от возрастания, а также соответствующие им широковещательные адреса подсетей

Этап 2 Обратите внимание на начало и конец диапазона адресов, отметив самый низкий (в цифровой форме) идентификатор подсети и самый высокий (в цифровой форме) широковещательный адрес подсети

Этап 3 Выберите начальную длину префикса /P для этапа 4 следующим образом: самый короткий префикс маски всех зависимых подсетей минус 1

Этап 4 Используйте самый низкий идентификатор зависимой подсети и текущую длину префикса для вычисления нового идентификатора и широковещательного адреса подсети.

A. Если полученный диапазон включает весь диапазон из этапа 2, то наилучший суммарный маршрут найден.

B. В противном случае вычтите 1 из длины префикса и повторите этап 4

Как обычно, сами этапы устрашают. Вот более короткая версия: выберите самый низкий идентификатор подсети из списка, продолжайте сокращать самую короткую маску в префиксном стиле, вычисляйте на их основании новый идентификатор подсети и смотрите, включает ли новая подсеть все адреса из исходной подсети. Но лучше всего действительно понять, что и зачем нужно.

Пример наилучшего суммарного маршрута для маршрутизатора R3

Маршрутизатор R3 на рис. 21.1 и 21.2 выше был подключен к подсетям 10.3.4.0/24, 10.3.5.0/24, 10.3.6.0/24 и 10.3.7.0/24. На рис. 21.3 приведены результаты первых трех следующих этапов процесса применительно к трем маршрутам от маршрутизатора R3.

Этап 1 Повторно перечислите идентификаторы подсетей (и длины префикса), а затем вычислите широковещательные адреса подсетей

Этап 2 Выберите идентификатор 10.3.4.0 как самый меньший идентификатор подсети и 10.3.7.255 как самый больший широковещательный адрес подсети, определив нижний и верхний пределы диапазона

Этап 3 При всех четырех масках /24 вычтите 1 из 24 и получите исходное значение /P, равное /23

Рис. 21.3. Поиск наилучшего суммарного маршрута (первые три этапа первого примера)

В данном случае этап 4 начинается с использования идентификатора подсети 10.3.4.0 и маски /23. На настоящий момент даже неизвестно, будет ли 10.3.4.0 номером подсети при использовании маски /23, поэтому проведите вычисления, как будто пытаетесь найти номер подсети и широковещательный адрес. Вычисления дают такой результат:

/23: подсеть 10.3.4.0, широковещательный адрес 10.3.5.255.

На этапе 4A сравниваются вновь вычисленный диапазон адресов подсети с диапазоном адресов в первоначальных подсетях, выявленным на этапе 2. Новый потенциально лучший суммарный маршрут не включает весь диапазон адресов первоначальных подсетей. Таким образом, на этапе 4B вычитаем 1 из длины префикса (23 – 1 = 22) и повторяем этап 4, но уже с маской /22.

Снова начинаем этап 4 с самого низкого первоначального идентификатора подсети (10.3.4.0) и, используя текущий префикс /22, вычисляем идентификатор и широковещательный адрес подсети. Получаем:

/22: подсеть 10.3.4.0, широковещательный адрес 10.3.7.255.

Вернувшись к этапу 4A, выясняем, что этот диапазон точно соответствует диапазону, представленному на рис. 21.3. Таким образом, найдены подсеть и маска для использования суммарным маршрутом: 10.3.4.0/22.

Пример наилучшего суммарного маршрута для маршрутизатора R2

На рис. 21.1 показаны четыре подсети справа и четыре подсети слева. До сих пор в данной главе игнорировались подсети слева, но теперь можно вычислить наилучший суммарный маршрут и для них. Их маршруты: 10.2.1.0/24, 10.2.2.0/24, 10.2.3.0/24 и 10.2.4.0/24.

Результаты первых трех этапов представлены на рис. 21.4.

Этап 1 Повторно перечислите идентификаторы подсетей (и длины префикса), а затем вычислите широковещательные адреса подсетей

Этап 2 Выберите идентификатор 10.2.1.0 как самый меньший идентификатор подсети и 10.2.4.255 как самый больший широковещательный адрес подсети, определив нижний и верхний пределы диапазона

Этап 3 При всех четырех масках /24, вычтите 1 из 24 и получите исходное значение /P, равное /23

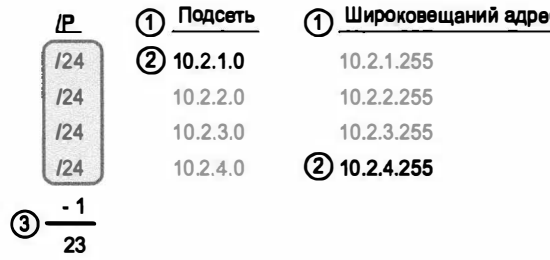


Рис. 21.4. Поиск наилучшего суммарного маршрута (первые три этапа второго примера)

Вначале этапа 4 используются идентификатор подсети 10.2.1.0 и маска /23. На настоящий момент даже неизвестно, будет ли 10.2.1.0 номером подсети при использовании маски /23, поэтому проведите вычисления, как будто пытаетесь найти номер подсети и широковещательный адрес. Вычисления дают такой результат:

/23: подсеть 10.2.0.0, широковещательный адрес 10.2.1.255.

На этапе 4А, сравниваются вновь вычисленный диапазон адресов подсети с диапазоном на рис. 21.4. Новый потенциально лучший суммарный маршрут не включает весь диапазон адресов первоначальных подсетей. Таким образом, на этапе 4Б вычитаем 1 из длины префикса (23 – 1 = 22) и повторяем этап 4, но уже с маской /22.

Снова начинаем этап 4 с самого низкого первоначального идентификатора подсети (10.2.1.0) и, используя текущий префикс /22, вычисляем идентификатор и широковещательный адрес подсети. Получаем:

/22: подсеть 10.2.0.0, широковещательный адрес 10.2.3.255.

Этот новый диапазон включает адреса трех из четырех первоначальных зависимых подсетей, но не подсети 10.2.4.0/24. Поэтому нужно еще раз повторить этап 4, на сей раз с маской /21, что дает следующий результат:

/21: подсеть 10.2.0.0, широковещательный адрес 10.2.7.255.

Новая подсеть включает весь диапазон, поэтому это наилучший суммарный маршрут для данных подсетей.

Практические задания по выбору наилучшего суммарного маршрута

В табл. 21.1 приведены четыре набора подсетей, которые должны войти в суммарный маршрут. Найдите комбинацию номера подсети и маски, являющуюся наилучшим суммарным маршрутом, согласно определению, данному в предыдущем разделе.

Таблица 21.1. Практические задания: поиск наилучшего суммарного маршрута

Задание 1	Задание 2	Задание 3	Задание 4
10.1.50.0/23	172.16.112.0/24	192.168.1.160/28	172.16.125.0/24
10.1.48.0/23	172.16.114.0/25	192.168.1.152/30	172.16.126.0/24
10.1.46.0/23	172.16.116.0/23	192.168.1.192/29	172.16.127.0/24
10.1.52.0/23	172.16.111.0/24	192.168.1.128/28	172.16.128.0/24

Ответы приведены в конце главы, в разделе “Ответы на практические задания главы”.

Обзор

Резюме

- Инструментальные средства суммирования маршрутов позволяют инженерам анонсировать один маршрут, который заменяет несколько других маршрутов новым, соответствующим тому же диапазону адресов. Это позволяет снизить нагрузку и сэкономить ресурсы процессора, полосу пропускания, а также не растрачивать память впустую.
- У суммирования маршрутов есть много преимуществ. Оно сокращает размер таблиц маршрутизации, тем не менее позволяя маршрутизатору перенаправлять пакеты ко всем получателям в сети. Более короткая таблица позволяет повысить производительность маршрутизации и сэкономить память на каждом маршрутизаторе. Суммирование улучшает также время конвергенции для протоколов маршрутизации, поскольку у них существенно снизится объем работы.
- Суммирование маршрутов вручную заставит маршрутизатор прекратить анонсировать набор маршрутов, анонсируя вместо них единый маршрут, содержащий набор всех адресов. Для этого маршрутизатор, создающий суммарный маршрут, следует настроить так, чтобы он знал анонсируемый номер подсети и маску.
- Под наилучшим суммарным маршрутом в этой книге понимается следующее: *Суммарный маршрут с наименьшим диапазоном адресов, который включает все адреса всех подсетей, которые необходимо объединить в один суммарный маршрут.*
- Ниже описаны этапы поиска наилучшего суммарного маршрута.

  Этап 1 Перечислите все номера подлежащих суммированию (зависимых) подсетей в десятичном виде в порядке возрастания, а также соответствующие им широковещательные адреса подсетей

  Этап 2 Обратите внимание на начало и конец диапазона адресов, отметив самый низкий (в цифровой форме) идентификатор подсети и самый высокий (в цифровой форме) широковещательный адрес подсети

  Этап 3 Выберите начальную длину префикса /P для этапа 4 следующим образом: самый короткий префикс маски всех зависимых подсетей минус 1

  Этап 4 Используйте самый низкий идентификатор зависимой подсети и текущую длину префикса для вычисления нового идентификатора и широковещательного адреса подсети.

  A. Если полученный диапазон включает весь диапазон из этапа 2, то наилучший суммарный маршрут найден.

  B. В противном случае вычтите 1 из длины префикса и повторите этап 4

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Какая из следующих просуммированных подсетей представляет собой наименьший суммарный маршрут (имеет наименьший диапазон адресов), который включает подсети 10.3.95.0, 10.3.96.0 и 10.3.97.0 с маской 255.255.255.0?

   А) 10.0.0.0 255.0.0.0.
   Б) 10.3.0.0 255.255.0.0.
   В) 10.3.64.0 255.255.192.0.
   Г) 10.3.64.0 255.255.224.0.

2. Какая из указанных ниже просуммированных подсетей не является допустимым суммарным маршрутом, который включает подсети 10.1.55.0, 10.1.56.0 и 10.1.57.0 с маской 255.255.255.0? (Выберите два ответа.)

   А) 10.0.0.0 255.0.0.0.
   Б) 10.1.0.0 255.255.0.0.
   В) 10.1.55.0 255.255.255.0.
   Г) 10.1.48.0 255.255.248.0.
   Д) 10.1.32.0 255.255.224.0.

3. Что из следующего было бы наилучшей подсетью и маской для нового суммарного маршрута, суммирующего маршруты к подсетям 10.1.12.0/24, 10.1.14.0/24 и 10.1.15.0/24?

   А) 10.1.0.0/20.
   Б) 10.1.8.0/21.
   В) 10.1.12.0/21.
   Г) 10.1.12.0/22.

4. Что из следующего было бы наилучшей подсетью и маской для нового суммарного маршрута, суммирующего маршруты к подсетям 192.168.1.64/28, 192.168.1.80/28 и 192.168.1.96/28?

   А) 192.168.1.0/25.
   Б) 192.168.1.64/26.
   В) 192.168.1.32/26.
   Г) 192.168.1.64/27.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. ниже.

Таблица 21.2. Ключевые темы главы 21

Элемент	Описание	Страница
Определение	Критерии, определяющие суммарный маршрут как наилучший для данного набора подсетей	624
Список	Процесс поиска наилучшего маршрута при суммировании вручную	624

Практические задания в приложении И

Дополнительные практические задания и ответы на них приведены в приложении И на веб-странице книги в формате PDF.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

классовый протокол маршрутизации (classful routing protocol), бесклассовый протокол маршрутизации (classless routing protocol), перекрывающиеся подсети (overlapping subnets), маски подсети переменной длины (Variable-Length Subnet Masks — VLSM), неразрывная сеть (contiguous network), изолированная подсеть (discontiguous network), суммарный маршрут (summary route).

Практика

Ответы на практические задания главы

В этом разделе приведены ответы на пять практических заданий раздела "Практические задания по выбору наилучшего суммарного маршрута". Ответы приведены вместе с описанием использования процесса решения задачи.

В табл. 21.3 приведены результаты первых двух этапов для каждой задачи; поля, выделенные серым, демонстрируют верхнюю и нижнюю границы диапазона, который должен включать новый суммарный маршрут. В табл. 21.4 демонстрируются результаты каждого прохода через этап 4, с окончательным (справа) проходом, содержащим правильный ответ.

Задание 1

Таблица 21.3. Практическое задание 1: первые два этапа

Идентификатор подсети и маска	Широковещательный адрес подсети
10.1.50.0/23	10.1.51.255
10.1.48.0/23	10.1.49.255
10.1.46.0/23	10.1.47.255
10.1.52.0/23	10.1.53.255

Для задания 1 на этапе 3 все маски — /23, поэтому начальная маска будет на 1 меньше, или /22. Поиск правильного ответа требует четырех проходов вычисления нового идентификатора подсети и маски. Окончательный ответ представлен в табл. 21.4.

Таблица 21.4. Практическое задание 1: несколько проходов выполнения этапа 4 (правильный ответ выделен)

Везде используется 10.1.46.0	1 проход: /22	2 проход: /21	3 проход: /20	4 проход: /19
Идентификатор подсети	10.1.44.0	10.1.40.0	10.1.32.0	10.1.32.0
Широковещательный адрес	10.1.47.255	10.1.47.255	10.1.47.255	10.1.63.255

Задание 2

Таблица 21.5. Практическое задание 2: первые два этапа

Идентификатор подсети и маска	Широковещательный адрес подсети
172.16.112.0/24	172.16.112.255
172.16.114.0/25	172.16.114.127
172.16.116.0/23	172.16.117.255
172.16.111.0/24	172.16.111.255

Для задания 2 на этапе 3 самая короткая маска — /23, начальная маска будет на 1 меньше, или /22. Поиск правильного ответа требует четырех проходов вычисления нового идентификатора подсети и маски. Окончательный ответ приведен в табл. 21.6.

Таблица 21.6. Практическое задание 2: несколько проходов выполнения этапа 4 (правильный ответ выделен)

Везде используется 172.16.111.0	1 проход: /22	2 проход: /21	3 проход: /20	4 проход: /19
Идентификатор подсети	172.16.108.0	172.16.104.0	172.16.96.0	172.16.96.0
Широковещательный адрес	172.16.111.255	172.16.111.255	172.16.111.255	172.16.127.255

Задание 3

Таблица 21.7. Практическое задание 3: первые два этапа

Идентификатор подсети и маска	Широковещательный адрес подсети
192.168.1.160/28	192.168.1.175
192.168.1.152/30	192.168.1.155
192.168.1.192/29	192.168.1.199
192.168.1.128/28	192.168.1.143

Для задания 3 на этапе 3 самая короткая маска — /28, поэтому начальная маска будет на 1 меньше, или /27. Поиск правильного ответа требует трех проходов вычисления нового идентификатора подсети и маски. Окончательный ответ представлен в табл. 21.8.

Таблица 21.8. Практическое задание 3: несколько проходов выполнения этапа 4 (правильный ответ выделен)

Везде используется 192.168.1.128	1 проход: /27	2 проход: /26	3 проход: /25
Идентификатор подсети	192.168.1.128	192.168.1.128	192.168.1.128
Широковещательный адрес	192.168.1.159	192.168.1.191	192.168.1.255

Задание 4

Таблица 21.9. Практическая задача 4: первые два этапа

Идентификатор подсети и маска	Широковещательный адрес подсети
172.16.125.0/24	172.16.125.255
172.16.126.0/24	172.16.126.255
172.16.127.0/24	172.16.127.255
172.16.128.0/24	172.16.128.255

Для задания 4 на этапе 3 самая короткая маска — /24, поэтому начальная маска будет на 1 меньше, или /23.

Таблица 21.10. Практическое задание 4: несколько проходов выполнения этапа 4 (правильный ответ выделен)

Везде используется 172.16.125.0	1 проход: /23	2 проход: /22	3 проход: /21	4 проход: /20
Идентификатор подсети	172.16.124.0	172.16.124.0	172.16.120.0	172.16.112.0
Широковещательный адрес	172.16.125.255	172.16.127.255	172.16.127.255	172.16.127.255

В табл. 21.10 все же нет правильного ответа. Процесс придется продолжать до маски /16, прежде чем будет найден наилучший суммарный маршрут 172.16.0.0/16.

Ответы на контрольные вопросы:

1 В. **2** В и Г. **3** Г. **4** Б.

Обзор части V

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части V

Задание	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей процесса		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав только этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения к книге.

Ответы на вопросы

Ответьте на вопросы обзора этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров только этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения к книге.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если понятны не все их подробности, уделите время повторному изучению.

Создайте диаграмму связей процесса

В этой части рассматривается несколько типов проблем, решаемых способами, изложенными в данной главе. Следующая диаграмма связей поможет повторить большинство концепций по каждому типу проблем. Этот обзор не сосредоточивается на подробностях поиска ответа на каждую задачу — для этого есть практические занятия в конце глав 19–21.

В этих главах рассматриваются четыре основных типа задач, которые могут быть решены арифметически.

- *Выбор масок подсети*. Выберите единую маску для использования во всей классовой сети IP на основании требований проекта.
- *Поиск всех идентификаторов подсети*. Вычислите все идентификаторы подсети в сети.

- *Поиск перекрытий масок VLSM.* Обнаружение ошибок в проекте, где перекрываются интервалы адресов двух или более подсетей.
- *Добавление новых подсетей к существующему проекту VLSM.* Найдите свободную область в существующем проекте подсетей для добавления новой подсети VLSM.
- *Поиск наилучшего суммарного маршрута.* На основании списка подсетей и масок найдите объединяющий их суммарный маршрут, но с наименьшим количеством дополнительных адресов.

Создайте диаграмму связей с ветвями по каждой теме в списке. Начните каждую ветвь с базовой концепции и разбейте ее на три подраздела, как показано на рис. Ч5.1.

- *Дано* (given). Имеющаяся и подразумеваемая информация, на основании которой предстоит решить задачу.
- *Процесс* (process). Информация или термины, используемые во время процесса. Не описывайте отдельные этапы процесса; задача здесь в том, чтобы вспомнить, что это именно этот процесс, а не другой.
- *Результат* (result). Факты, определяемые при решении задачи.

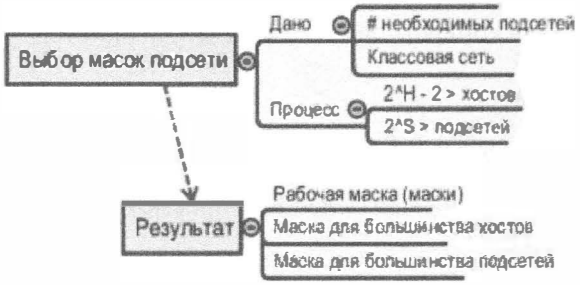


Рис. Ч5.1 Пример диаграммы связей для задач части V

ВНИМАНИЕ!

Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения к данной книге.

Если решено использовать программное обеспечение диаграмм связей, а не лист бумаги, имеет смысл запомнить место сохранения файлов диаграмм связей; в таблице ниже перечислены диаграммы связей для данной части, имена их файлов и места сохранения.

Диаграммы связей обзора части V

Диаграмма	Описание	Где сохранен результат
1	Диаграмма связей процесса	

Ответы приведены в приложении П (Appendix O) на веб-сайте, но ваши диаграммы связей могут выглядеть иначе.

До сих пор в этой книге рассматривались основы создания сетей IPv4 с использованием маршрутизаторов, коммутаторов, локальных сетей Ethernet и последовательных каналов связи. Часть VI завершает тему сетей IPv4 в этой книге рассмотрением некоторых служб, помогающих обеспечить защиту корпоративных сетей, а также полезных возможностей адресации при подключении к Интернету.

В главах 22 и 23 обсуждаются основы и дополнительные возможности списков управления доступом IPv4 (ACL). Списки ACL — это настраиваемые фильтры пакетов IPv4, просматривающие заголовки пакетов IPv4 и позволяющие пакету пройти или отбрасывающие его. В главе 23 рассматриваются некоторые другие темы защиты сети в дополнение к спискам ACL. В последней главе этой части, главе 24, описана *трансляция сетевых адресов* (NAT). Трансляция NAT помогает решить наибольшую проблему IPv4-адресации в Интернете и используется практически каждым домашним и корпоративным пользователем Интернета.

Часть VI. Службы IPv4

ГЛАВА 22

Простые списки управления доступом IPv4

Большинство тем экзамена CCENT сосредоточено на достижении основной задачи любой сети TCP/IP: доставке пакетов IPv4 с хоста отправителя на хост получателя. Эта, а также следующая глава сосредоточена на прямо противоположном: как с помощью списков управления доступом IPv4 (ACL) позволить только некоему подмножеству пакетов достигать получателей.

У списков ACL IP есть множество областей применения, но экзамен CCENT сосредоточивается на наиболее известном: фильтрации пакетов. Хосты в подсети, безусловно, должны быть в состоянии общаться с хостами по всей корпоративной сети, но, возможно, есть группа серверов с секретными данными, которые следует защитить. Возможно, законодательство требует обеспечить защиту доступа не только по имени пользователя и паролю, но и контролировать также способность доставить пакет защищенному хосту или серверу. Списки ACL IP обеспечивают решение этих задач.

В этой главе обсуждаются основы списков ACL IP, в частности, один из типов списков ACL IP: стандартные нумерованные списки управления доступом IPv4. Глава 23 завершает обсуждение описанием других типов списков ACL IP.

ВНИМАНИЕ!

Хотя существуют также списки ACL IPv6, они не рассматриваются в этой книге, ни в первом, ни во втором томе. Все упоминания списков ACL IP в этой главе относятся именно к спискам ACL IPv4.

В этой главе рассматриваются следующие экзаменационные темы

Службы IP

- Типы, средства и приложения ACL.
  - Стандартные.
  - Нумерованные.
  - Средства регистрации.
- Настройка и проверка ACL в сетевой среде.
- Нумерованные.
- Средства регистрации.

Защита сетевых устройств

- Настройка и проверка списков ACL для фильтрации сетевого трафика.

Поиск и устранение неисправностей

- Поиск неисправностей и решение проблем списков ACL.
- Статистика.
- Разрешенные сети.
- Направление.
- Интерфейс.

Основные темы

Основы списков управления доступом IPv4

Списки управления доступом (Access Control List — ACL) IPv4 позволяют сетевым инженерам идентифицировать пакеты различных типов. Для этого в конфигурации ACL перечисляют значения, которые маршрутизатор может найти в заголовках IP, TCP, UDP и др. Например, список ACL может распознать пакет с IP-адресом отправителя 1.1.1.1, или пакеты, IP-адрес получателя которых находится в подсети 10.1.1.0/24, или пакеты с портом получателя TCP 23 (Telnet).

Списки ACL IPv4 выполняют множество функций в маршрутизаторах Cisco, но чаще всего они используются как фильтр пакетов. Инженеры могут применить списки ACL на маршрутизаторе, чтобы они контролировали передачу пакетов, проходящих через маршрутизатор. Теперь для каждого пакета IP маршрутизатор решает, отбросить или разрешить его дальнейшую передачу, как будто списков ACL не существовало.

Однако списки ACL применимы и для многих других задач IOS. Так, например, списки ACL применяются для распознавания пакетов средствами оценки *качества обслуживания* (Quality of Service — QoS). QoS позволяет маршрутизатору предоставлять одним пакетам лучшее обслуживание, чем другим. Например, пакеты, содержащие оцифрованный голос, должны иметь очень низкую задержку, поэтому списки ACL с логикой QoS позволяют передавать голосовые пакеты быстрее, чем пакеты данных.

В первом разделе представлено введение в списки ACL, используемые для фильтрации пакетов, и рассмотрены три их аспекта: расположение и контролируемое направление передачи, распознавание пакетов по их заголовкам и применение действий к распознанным пакетам.

Расположение и направление списков ACL

Маршрутизаторы Cisco способны применить логику ACL к пакетам в том месте, где пакеты IP поступают на интерфейс, или в том, где они покидают его. Другими словами, списки ACL ассоциируются с интерфейсом и направлением потока передачи пакетов (входящим или исходящим). Таким образом, список ACL может быть применен к входящим на маршрутизатор пакетам прежде, чем он примет решение об их перенаправлении, или к исходящим пакетам, после того, как маршрутизатор примет решение о перенаправлении и направит пакет на данный интерфейс.

Стрелки на рис. 22.1 показывают направления, в которых можно фильтровать пакеты (в текущей топологии слева направо). Предположим, например, что решено позволить передавать пакеты, посланные хостом A на сервер S1, но отбрасывать пакеты, посылаемые хостом B на сервер S1. Каждая отмеченная стрелками линия представляет возможные места и направления применения маршрутизатором списков ACL для фильтрации пакетов, посланных хостом B.

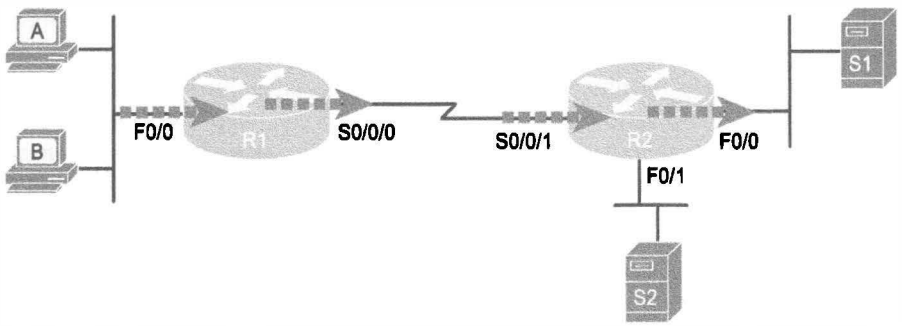


Рис. 22.1. Места для фильтрации пакетов, следующих от хостов A и B на сервер S1

Четыре линии со стрелками на рисунке указывают на расположение и направление интерфейсов маршрутизатора, используемых для перенаправления пакетов от хоста B на сервер S1. В данном конкретном примере этими интерфейсами и направлениями являются входящий интерфейс F0/0 маршрутизатора R1, исходящий интерфейс S0/0/0 маршрутизатора R1, входящий интерфейс S0/0/1 маршрутизатора R2 и исходящий интерфейс F0/0 маршрутизатора R2. Если, например, применить списки ACL на интерфейсе F0/1 маршрутизатора R2 в любом направлении, то этот список не будет фильтровать пакеты, посылаемые хостом B на сервер S1, поскольку интерфейс F0/1 маршрутизатора R2 не является частью маршрута от хоста B на сервер S1.

Общие правила расположения и направления для списков ACL

Ключевая тема

Короче говоря, чтобы фильтровать пакеты, необходимо разрешить применение списков ACL на интерфейсе, который обрабатывает пакет, в том же направлении, в котором пакеты следуют через этот интерфейс.

Когда список ACL применен, маршрутизатор обрабатывает каждый входящий или исходящий пакет IP, используя этот список. Например, если на интерфейсе F0/0 маршрутизатора R1 применен список ACL для входящих пакетов, то он будет сравнивать со списком ACL каждый входящий на интерфейс F0/0 пакет IP, чтобы решить, продолжать его перенаправление или отбросить пакет.

Распознавание пакетов

Обдумывая расположение и направление для списков ACL, следует учитывать также то, какие пакеты планируется отфильтровывать (отбрасывать), а какие пропускать. Чтобы довести эти идеи до маршрутизатора, на нем следует настроить список ACL IP, который распознает пакеты. *Распознавание пакетов* (matching packets) — это настраиваемый командами ACL процесс идентификации каждого пакета и принятие решения согласно списку, следует ли от него отказаться или позволить пройти далее.

Каждый список ACL IP состоит из одной или нескольких команд конфигурации, каждая из которых содержит список значений, искомых в заголовках пакета. Как правило, команда ACL использует следующую логику: "искать указанные значения в заголовке пакета, а если они найдены, отказаться от пакета". (Действием может быть разрешение на прохождение пакета вместо запрета.) Конкретно список ACL

задает поиск полей заголовка, которые должны быть уже знакомы, включая IP-адреса отправителя и получателя, а также номера портов TCP и UDP.

Рассмотрим пример на рис. 22.2, где необходимо разрешить передачу пакетов с хоста A на сервер S1, но запретить передачу на тот же сервер пакетов с хоста B. Теперь на рисунке представлены IP-адреса всех хостов, а также псевдокод списка ACL на маршрутизаторе R2. На рис. 22.2 представлено также выбранное расположение списка ACL: входящий интерфейс S0/0/1 маршрутизатора R2.

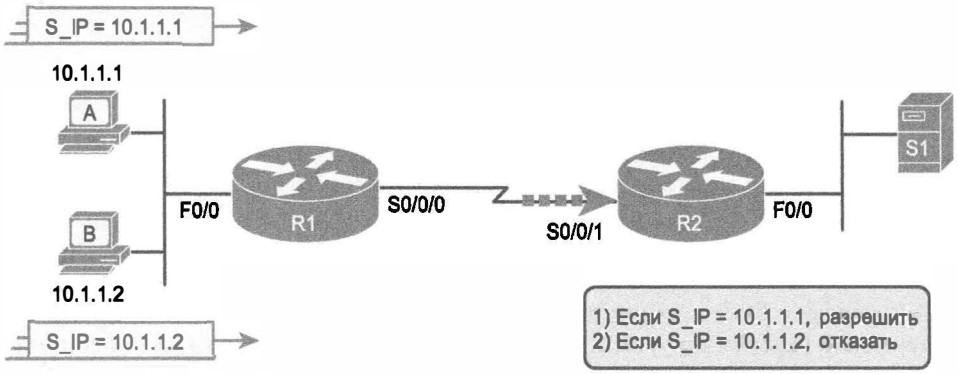


Рис. 22.2. Псевдокод, демонстрирующий логику команд распознавания списка ACL

На рис. 22.2 в прямоугольнике представлен список ACL из двух строк с простой логикой распознавания: оба оператора ищут соответствие только с IP-адресом отправителя пакета. При его применении маршрутизатор R2 просматривает каждый входящий пакет IP на этом интерфейсе и сравнивает IP-адрес его отправителя с тем, что указано в двух командах ACL. Пакеты, посланные хостом A (IP-адрес отправителя 10.1.1.1), проходят, а посланные хостом B (IP-адрес отправителя 10.1.1.2) блокируются.

Применение действий при найденном соответствии

При использовании списков ACL IP для фильтрации пакетов может быть выбрано только одно из двух действий. Команды конфигурации используют ключевые слова `deny` (запретить) и `permit` (разрешить), означающие (соответственно) отказ от пакета или разрешение его передачи, как будто список ACL не существует.

Эта книга сосредоточивается на использовании списков ACL для фильтрации пакетов, но операционная система IOS применяет их еще и для других целей. Там, как правило, используется та же логика распознавания. Однако в других случаях ключевые слова `deny` и `permit` подразумевают некое другое действие. Например, в главе 24 списки ACL используются для распознавания пакетов, но ключевое слово `permit` указывает маршрутизатору применять функции NAT, которые преобразуют IP-адреса.

Типы списков ACL IP

Операционная система Cisco IOS поддерживала списки ACL IP с самых первых маршрутизаторов Cisco. Начиная с первоначальных стандартных нумерованных

списков ACL IP первых версий операционной системы IOS, способных обеспечить логику, представленную на рис. 22.2, впоследствии компания Cisco добавила в списки ACL множество нововведений, включая следующие.

- Стандартный нумерованный список ACL (1–99).
- Расширенный нумерованный список ACL (100–199).
- Дополнительные номера ACL (стандартные 1300–1999, расширенные 2000–2699).
- Именованные списки ACL.
- Улучшенное редактирование с помощью порядковых номеров.

Основное внимание в этой главе уделяется исключительно стандартным нумерованным спискам ACL IP, а в главе 23 обсуждаются три других основных категории списков ACL IP. Короче говоря, списки ACL IP бывают нумерованными или именованными, в конфигурации которых используются либо номера, либо имена. Списки ACL бываю также стандартными или расширенными, последние имеют много больше возможностей распознавания пакетов. На рис. 22.3 приведены общие концепции, связанные с категориями списков ACL IP.

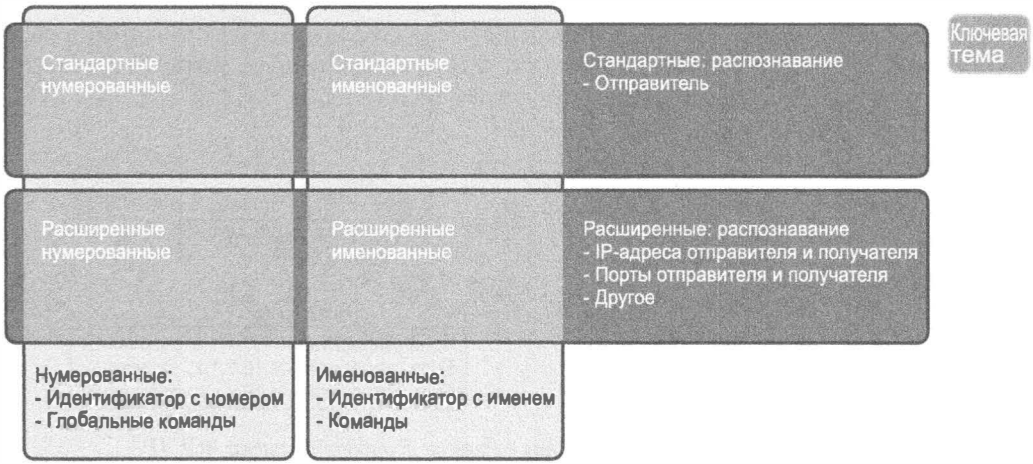


Рис. 22.3. Четыре основные категории списков ACL IPv4 операционной системы Cisco IOS

Стандартные нумерованные списки ACL IPv4

Заголовок этого раздела уже служит неплохим введением, если понимаешь, что именно Cisco подразумевает под каждым словом. Этот раздел о типе фильтра Cisco (ACL), который распознает только IP-адрес отправителя пакета (*стандартный*), настроенный на идентификацию списков ACL по номерам (*нумерованный*), а не по именам и проверяющий пакеты протокола IPv4.

В данном разделе рассматриваются, в частности, стандартные нумерованные списки ACL IP. Сначала представлена идея того, что ACL — это список, а также какую логику он использует. Далее подробно рассматривается распознавание поля IP-адреса отправителя в заголовке пакета, включая синтаксис команд. Завершается

раздел полным обзором команд конфигурации и проверки, использующихся при реализации стандартных списков управления доступом.

Логика списков ACL IP

Один список ACL — это единая сущность, и в то же время это список из одной или нескольких команд конфигурации. Как единая сущность, весь список ACL применим на интерфейсе в определенном направлении, как было показано на рис. 22.1. Поскольку это список команд, у каждой команды собственная логика распознавания, которую маршрутизатор должен применять к каждому пакету при фильтрации с использованием данного списка ACL.

В процессе обработки списка ACL маршрутизатор исследует пакет, сравнивая его со списком ACL, следующим образом.

Ключевая тема **Логика первого соответствия используется всеми списками ACL**

Списки ACL используют логику первого соответствия. Как только обнаруживается соответствие пакета одной из строк списка ACL, маршрутизатор предпринимает действие, указанное в этой строке списка, и прекращает дальнейшее сравнение.

Для демонстрации того, что это означает, рассмотрим пример, представленный на рис. 22.4. На рисунке приведен пример списка ACL 1 с тремя строками псевдокода. Данный пример применяет список ACL 1 на входящем интерфейсе S0/0/1 маршрутизатора R2 (то же расположение, что и на рис. 22.2).

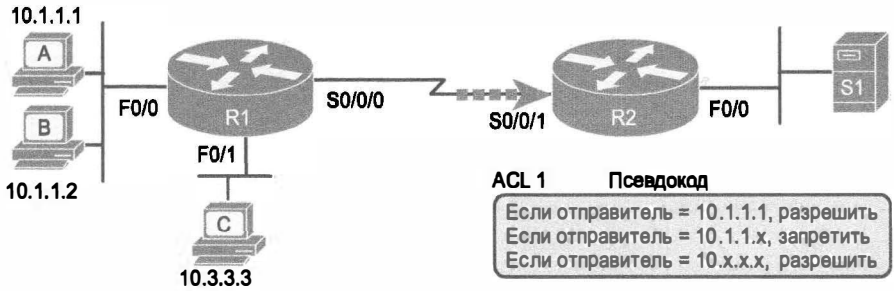


Рис. 22.4. Пример для обсуждения процесса обработки списка ACL IP

Рассмотрим логику первого соответствия списка ACL для пакета, посланного хостом A на сервер S1. IP-адресом отправителя будет 10.1.1.1, и он будет направлен на интерфейс S0/0/1 маршрутизатора R2, а значит, подпадает под действие логики списка ACL 1. Маршрутизатор R2 сравнивает этот пакет со списком ACL и находит первый соответствующий элемент, содержащий действие разрешения. Таким образом, этому пакету будет позволено пройти, как показано на рис. 22.5, *слева*.

Теперь рассмотрим пакет, посланный хостом B, с IP-адресом отправителя 10.1.1.2. Когда пакет поступает на интерфейс S0/0/1 маршрутизатора R2, он сравнивает его с первым оператором списка ACL 1 и не находит соответствия (10.1.1.1 не равно 10.1.1.2). Затем маршрутизатор R2 переходит ко второму оператору, который требует некоторого разъяснения. В псевдокоде ACL на рис. 22.4 представлена строка 10.1.1.x, которая является сокращением для любого значения в последнем октете.

Сравнивая только первые три октета, маршрутизатор R2 обнаружит, что IP-адрес отправителя данного пакета действительно начинается с трех октетов 10.1.1. Таким образом, маршрутизатор R2 посчитает его соответствующим второму оператору и предпримет соответствующее действие, отбросив пакет. Маршрутизатор R2 остановит обработку списка ACL для пакета, игнорируя третью строку списка ACL.

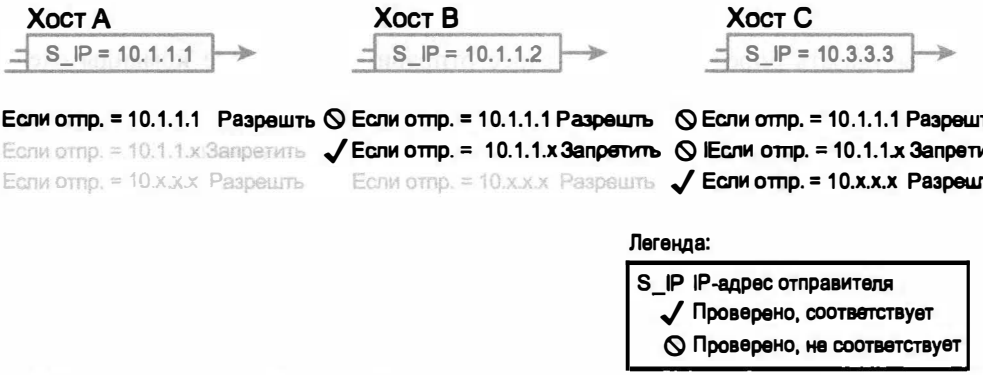


Рис. 22.5. Сравнение записей списка ACL с пакетом, передаваемым хостами A, B и C на рис. 22.4

И наконец, рассмотрим пакет, посланным хостом C также на сервер S1. IP-адрес отправителя пакета — 10.3.3.3, поэтому, попав на интерфейс S0/0/1 маршрутизатора R2, он проверяется по списку ACL. Маршрутизатор находит в списке ACL 1 первую команду, и оказывается, что она не соответствует пакету (10.1.1.1 в команде, не равно 10.3.3.3 у пакета). Затем маршрутизатор R2 исследует вторую команду, сравнивает первые три октета (10.1.1) с IP-адресом отправителя пакета (10.3.3) и снова не находит соответствия. Затем он просматривает третью команду. В данном случае шаблон предписывает игнорировать последние три октета и сравнивать только первый октет (10), в результате соответствие обнаруживается. Затем маршрутизатор R2 выясняет действие (разрешение) и позволяет пакету следовать далее.

Эта последовательность обработки списка ACL осуществляется для любого типа списков ACL операционной системы IOS: IP, других протоколов, стандартных или расширенных, именованных или нумерованных.

И наконец, если пакет не соответствует ни одной из записей списка ACL, то он отбрасывается. Дело в том, что конец любого списка ACL IP считается оператором *deny all* (запретить все). Он не указывается в конфигурации, но если маршрутизатор не нашел соответствия до конца списка, то операционная система IOS полагает, что пакет соответствует записи с действием `deny`.

Логики распознавания и синтаксис команд

Стандартные нумерованные списки ACL IP используют следующую глобальную команду:

`access-list {1-99 | 1300-1999} {permit|deny}` *`параметры_соответствия`*

Каждый стандартный нумерованный список ACL содержит одну или несколько команд `access-list` с любым номером из диапазона, представленного в строке синтаксиса выше. (Ни один номер не лучше другого.)

Помимо номера ACL, каждая команда `access-list` содержит также выбранное действие (`permit` или `deny`) и логику распознавания. В оставшейся части этого раздела рассматривается настройка параметров распознавания, которыми для стандартных списков доступа может быть только распознавание IP-адреса отправителя или его части с использованием *шаблона маски* (wildcard mask).

Точное распознавание IP-адреса

Чтобы указать определенный IP-адрес отправителя, в конце команды следует указать весь IP-адрес. Например, в предыдущем примере используется псевдокод "разрешить, если IP-адрес отправителя = 10.1.1.1". Следующая команда задает логику точного соответствия для использования списка ACL номер 1:

```
access-list 1 permit 10.1.1.1
```

Точное распознавание IP-адреса просто.

В прежних, более ранних версиях операционной системы IOS синтаксис включал ключевое слово `host`. Вместо того чтобы просто ввести полный IP-адрес, сначала необходимо было ввести ключевое слово `host`, а затем IP-адрес. Обратите внимание: если использовать ключевое слово `host` в более поздних версиях операционной системы IOS, то она примет команду, а затем удалит ключевое слово:

```
access-list 1 permit host 10.1.1.1
```

Распознавание подмножества адресов по шаблону

Зачастую бизнес-задачи требуют реализации списков ACL, распознающих не один конкретный IP-адрес, а диапазон IP-адресов. Возможно, необходимо распознать все IP-адреса подсети, возможно, все IP-адреса диапазона подсетей, подобно группировке в суммарный маршрут, как было описано в предыдущей главе. Независимо от этого, вполне возможно распознавать несколько IP-адресов в диапазоне.

Операционная система IOS позволяет стандартным спискам доступа распознавать диапазон адресов, используя инструмент, называемый *шаблоном маски* (wildcard mask). Обратите внимание: это не маска подсети. Шаблон маски (который в этой книге сокращенно называется *маской WC*) позволяет инженеру указать операционной системе IOS игнорировать части адреса при сравнении; по существу, при применении шаблона эти части считаются совпадающими.

Маски WC можно рассматривать и в десятичном, и в двоичном виде, у каждого из них есть своя область применения. Для начала рассмотрим маски WC в десятичном виде, используя следующие правила.

Ключевая тема **Логика шаблона маски для десятичных чисел 0 и 255**

Десятичный 0. Маршрутизатор должен сравнивать этот октет, как обычно.

Десятичное 255. Маршрутизатор игнорирует этот октет, считая его совпадающим.

С учетом этих двух правил рассмотрим рис. 22.6, демонстрирующий эту логику на примере трех разных, но весьма популярных масок WC: предписывающую маршрутизатору игнорировать последний октет, последние два и последние три октета.

Рис. 22.6. Логика масок WC 0.0.0.255, 0.0.255.255 и 0.255.255.255

Все три примера в прямоугольниках на рис. 22.6 представляют два совершенно разных числа. Однако применение маски WC позволяет IOS сравнивать только некоторые из октетов, игнорируя другие. Все три примера дают соответствие, поскольку каждый шаблон маски указывает IOS игнорировать некоторые октеты. Пример слева демонстрирует маску WC 0.0.0.255, которая указывает маршрутизатору рассматривать последний октет как шаблон, по существу, игнорируя его при сравнении. Точно так же средний пример демонстрирует маску WC 0.0.255.255, которая указывает маршрутизатору игнорировать два октета справа. В случае, показанном слева, маска WC 0.255.255.255 указывает маршругизатору игнорировать при сравнении значений последние три октета.

Чтобы увидеть маску WC в действии, вернемся к более раннему примеру, связанному с рис. 22.4 и 22.5. Псевдокод списка ACL на этих двух рисунках использовал логику, которая может быть реализована с использованием масок WC. Напомним, что логика псевдокода ACL на этих рисунках была такова.

Логика шаблона маски при поиске соответствия подсети

Ключевая тема

- **Строка 1**. Соответствуют и разрешаются все пакеты с адресом отправителя точно 10.1.1.1.
- **Строка 2**. Соответствуют и запрещаются все пакеты с первыми тремя октетами 10.1.1 адреса отправителя.
- **Строка 3**. Соответствуют и разрешаются все адреса с первым октетом 10.

На рис. 22.7 приведена измененная версия рис. 22.4, но с полным правильным синтаксисом, включая маски WC. В частности, обратите внимание на использование маски WC 0.0.0.255 во второй команде, указывающей маршрутизатору R2 игнорировать последний октет числа 10.1.1.0, а маска WC 0.255.255.255 в третьей команде указывает маршругизатору R2 игнорировать последние три октета значения 10.0.0.0.

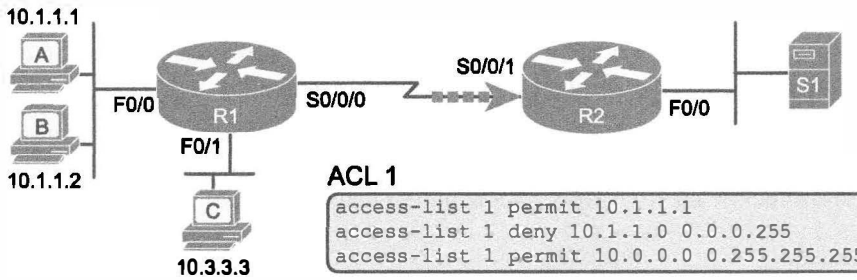


Рис. 22.7. Псевдокод, приведенный на рис. 22.4, заменен синтаксически правильным списком ACL

И наконец, обратите внимание на то, что при использовании маски WC все октеты IP-адреса отправителя в параметре команды `access-list` должны содержать значение 0 там, где маска WC содержит значение 255. Операционная система IOS желает, чтобы адрес отправителя содержал значение 0 в тех частях, которые будут проигнорированы, даже если заданы значения, отличные от нуля.

Двоичные шаблоны маски

Шаблоны маски, как значение в *десятичном представлении с разделительными точками* (Dotted-Decimal Notation — DDN), фактически представляют 32-разрядное двоичное число. Как 32-разрядное число, маска WC фактически поддерживает побитовую логику маршрутизатора. Короче говоря, бит 0 маски WC означает обычное сравнение, а двоичное значение 1 означает шаблон, т.е. позицию, которая при сравнении чисел игнорируется.

К счастью, на экзаменах CCENT и CCNA и, как правило, в большинстве реальных ситуаций двоичные маски WC можно игнорировать. Почему? Обычно необходимо распознать диапазон адресов, которые могут быть легко идентифицированы по номеру подсети и маске, или реальную подсеть, или суммарный маршрут, объединяющий подсети. (Более подробно о суммарных маршрутах см. в главе 21.) Если диапазон адресов можно описать с помощью номера подсети и маски, то с помощью простой математики, обсуждаемой далее, можно найти и числа, используемые в списке ACL.

ВНИМАНИЕ!

Если действительно хочется узнать логику двоичной маски, возьмите два числа DDN, которые сравнивают список ACL (одно из команды `access-list`, а второе из заголовка пакета), и преобразуйте их в двоичный формат. Затем преобразуйте также в двоичный формат маску WC. Сравните бит за битом первые два двоичных числа, игнорируя любые биты, для которых маска WC предоставляет двоичное значение 1, поскольку 1 указывает игнорировать этот бит. Если все проверенные биты равны, то значения соответствуют друг другу.

Поиск шаблона маски для соответствия подсети

Во многих случаях список управления доступом требует сопоставления со всеми хостами в конкретной подсети. Чтобы сопоставить подсеть со списком управления доступом, можно воспользоваться следующим упрощенным методом.

Ключевая тема **Этапы планирования и реализации стандартных списков ACL IP**

- Использовать номер подсети в качестве значения адреса отправителя в команде `access-list`.
- Использовать шаблон маски, найденный в результате вычитания маски подсети из точечно-десятичного значения 255.255.255.255.

Например, для подсети 172.16.8.0 с маской 255.255.252.0 следует использовать номер подсети (172.16.8.0) в качестве параметра адреса, а затем выполнить следующие математические вычисления, чтобы найти шаблон маски:

```
  255.255.255.255
- 255.255.252.0
-----------------
    0.  0.  3.255
```

Некоторые экзаменационные вопросы могут не требовать подготовки команд для списка управления доступом, который должен быть введен в конфигурацию, а вместо этого задание формулируется как требование интерпретировать некоторые существующие команды `access-list`. Как правило, в таких вопросах перечислены введенные в конфигурацию операторы списка управления доступом или предъявлено требование вывести содержимое списка управления доступом на имитаторе маршрутизатора, а экзаменуемый должен определить, какому оператору соответствует каждый конкретный пакет. Для этого необходимо определить диапазон IP-адресов, сопоставляемый с помощью каждой конкретной комбинации адреса и шаблона маски, в каждом операторе списка управления доступом.

Тот, кто мастерски владеет математическими вычислениями в области расчета подсетей на основе любого из упрощенных методов работы с десятичными представлениями и может обойтись без вычислений с двоичными представлениями, имеет возможность воспользоваться еще одним упрощенным методом, чтобы проанализировать каждую существующую пару адреса и шаблона маски в каждой команде списка управления доступом. Ниже описан соответствующий процесс.

Советы по созданию логики соответствия для поля адреса отправителя в команде списка управления доступом

Ключевая тема

Этап 1 Используйте адрес в команде `access-list` так, как если бы он представлял собой номер подсети

Этап 2 Примените число, найденное за счет вычитания шаблона маски из значения 255.255.255.255, в качестве маски подсети

Этап 3 Трактуйте значения, полученные на этапах 1 и 2, как номер подсети и маску подсети и найдите широковещательный адрес для подсети. Список управления доступом сопоставляется с диапазоном адресов от номера подсети до широковещательного адреса включительно

Диапазон адресов, полученный с помощью этого процесса, является тем же диапазоном адресов, который соответствует рассматриваемому списку управления доступом. Таким образом, если читатель уже умеет быстро находить диапазоны адресов любых подсетей, он может воспользоваться указанным процессом вместо математических вычислений на основе списков управления доступом, чтобы быстрее находить ответ на экзаменах. Например, если речь идет о команде `access-list 1 permit 172.16.200.0 0.0.7.255`, то необходимо в первую очередь перейти к рассмотрению значения 172.16.200.0 как номера подсети. Затем можно вычислить предполагаемую маску подсети 255.255.248.0 так, как показано ниже.

```
255.255.255.255
- 0.  0.  7.255
255.255.248.  0
```

Продолжая этот пример, получим следующую законченную команду для той же подсети:

```
access-list 1 permit 172.16.8.0 0.0.3.255
```

Далее, в разделе "Практические задания на применение стандартных списков ACL", будет возможность попрактиковаться в настройке соответствия подсетям для списков ACL.

Соответствие всем и любым адресам

В некоторых случаях необходимо, чтобы одна команда ACL соответствовала всем и любым пакетам, которые достигают этой точки списка ACL. Сначала необходимо узнать способ, как, используя ключевое слово `any`, добиться соответствия всем пакетам. Но еще важнее знать, когда необходимо соответствие всем пакетам.

Для соответствия всем пакетам в команде ACL следует использовать вместо адреса ключевое слово `any`. Например, чтобы разрешить передачу всех пакетов:

```
access-list 1 permit any
```

Так где и когда использовать такую команду? Помните, все списки ACL IP маршрутизаторов Cisco завершаются неявной командой `deny any`. Таким образом, если маршрутизатор, сравнив пакет со всеми записями списка ACL, не находит соответствия, то он отказывается от пакета. Хотите переопределить это стандартное поведение? Расположите в конце списка команду `permit any`.

Можно также явно указать в конце списка ACL команду, запрещающую весь трафик (например, `accesslist 1 deny any`). Но зачем, когда та же логика, так или иначе, уже находится в конце списка ACL? Команда ACL `show` отображает счетчики количества пакетов, соответствующих каждой команде, в списке ACL, но никакого счетчика для неявной команды `deny any` в конце списка ACL нет. Поэтому, если необходим счетчик количества пакетов, соответствующих команде `deny any` в конце списка ACL, ее следует указать в конфигурации явно.

Реализация стандартного списка ACL IP

В этой главе все этапы конфигурации уже были продемонстрированы по частям. Данный раздел резюмирует эти части в единый процесс конфигурации. Процесс задействует также команду `access-list`, обобщенный синтаксис которой имеет следующий вид:

`access-list` *номер-списка* {`deny` | `permit`} *отправитель* [*шаблон_маски-отправителя*]

Этапы реализации стандартного списка ACL IP

Ключевая тема

Этап 1 Определите, где будете размещать список управления доступом (маршрутизатор и интерфейс) и направленность (входящий или исходящий) в применяемом интерфейсе, следующим образом:

А. Стандартные списки управления доступом следует помещать со стороны получателя пакетов, чтобы их применение не приводило к непреднамеренному уничтожению пакетов, которые не должны быть отброшены.

Б. Стандартные списки управления доступом позволяют проводить проверку только IP-адреса отправителя в пакете, поэтому следует определить, какими будут IP-адреса отправителей в пакетах, для проверки которых применяется список управления доступом

Этап 2 Настройте одну или нескольких глобальных команд конфигурации `access-list` для создания списка управления доступом, руководствуясь указанными ниже соображениями.

А. Поиск в списке происходит последовательно, причем пакет обрабатывается по первому же совпадающему с его характеристиками правилу. Иными словами, если

параметры пакета совпали с одним из операторов `access-list`, поиск заканчивается, несмотря на то, что пакет мог бы совпасть и с одним из последующих операторов.

B. Стандартное действие в том случае, если пакет не соответствует ни одной из команд `access-list`, состоит в запрещении прохождения пакета (и его уничтожении)

Этап 3 Примените список управления доступом на выбранном интерфейсе маршрутизатора с учетом нужной направленности с использованием команды режима конфигурирования интерфейса `ip access-group number {in | out}`

В оставшейся части этого раздела приведено несколько примеров.

Пример 1. Стандартный нумерованный список ACL

В первом примере приведена конфигурация для тех же требований, представленных на рис. 22.4 и 22.5. Напомним их.

1. Применить список ACL на входящем интерфейсе S0/0/1 маршрутизатора R2.
2. Разрешить передачу пакетов от хоста A.
3. Запретить передачу пакетов от других хостов в подсети хоста A.
4. Разрешить передачу пакетов с любых других адресов в сети класса A 10.0.0.0.
5. Исходный пример не имел никаких указаний о том, что делать по умолчанию. Поэтому просто запретите весь другой трафик.

Пример 22.1 демонстрирует законченную конфигурацию, начиная с процесса настройки и завершая выводом команды `show running-config`.

Пример 22.1. Стандартный нумерованный список ACL. Конфигурация примера 1

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# access-list 1 permit 10.1.1.1
R2(config)# access-list 1 deny 10.1.1.0 0.0.0.255
R2(config)# access-list 1 permit 10.0.0.0 0.255.255.255
R2(config)# interface S0/0/1
R2(config-if)# ip access-group 1 in
R2(config-if)# ^Z
R2# show running-config
! Строки пропущены для краткости

access-list 1 permit 10.1.1.1
access-list 1 deny   10.1.1.0 0.0.0.255
access-list 1 permit 10.0.0.0 0.255.255.255
```

Сначала обратите пристальное внимание на процесс настройки конфигурации вверху примера. Как можно заметить, команда `access-list` не изменяет приглашение к вводу команд с глобального на приглашение режима конфигурации, поскольку команды `access-list` являются глобальными командами конфигурации. Затем сравните их с выводом команды `show running-config`: подробности идентичны тому, что было добавлено в режиме конфигурации. И наконец, обратите внимание на команду `ip access-group 1 in`, под интерфейсом S0/0/1 маршрутизатора R2, которая задействует логику ACL (и по расположению, и по направлению).

Пример 22.2 содержит вывод команды show для маршрутизатора R2, демонстрирующий информацию об этом списке ACL. Команда show ip access-lists выводит подробности только о списках ACL IPv4, в то время как команда show access-lists выводит подробности о списках ACL IPv4, а также других типов, например ACL IPv6.

Пример 22.2. Команды ACL show на маршрутизаторе R2

```
R2# show ip access-lists
Standard IP access list 1
    10 permit 10.1.1.1 (107 matches)
    20 deny   10.1.1.0, wildcard bits 0.0.0.255 (4 matches)
    30 permit 10.0.0.0, wildcard bits 0.255.255.255 (10 matches)
R2# show access-lists
Standard IP access list 1
    10 permit 10.1.1.1 (107 matches)
    20 deny   10.1.1.0, wildcard bits 0.0.0.255 (4 matches)
    30 permit 10.0.0.0, wildcard bits 0.255.255.255 (10 matches)
R2# show ip interface s0/0/1
Serial0/0/1 is up, line protocol is up
  Internet address is 10.1.2.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.9
  Outgoing access list is not set
  Inbound access list is 1
! Строки пропущены для краткости
```

Вывод этих команд демонстрирует два интересных момента. Первая строка вывода в данном случае отмечает тип (standard) и номер. Если бы существовало несколько списков ACL, то было бы несколько блоков вывода, по одному на каждый список ACL, со строкой заголовка, как здесь. Далее, эти команды перечисляют счетчики для количества пакетов, которые соответствуют каждой команде списка на маршрутизаторе. Например, первой строке в списке ACL на данный момент соответствует 107 пакетов.

В конце приведен вывод команды show ip interfaces. Эта команда отображает много других элементов, номера или имена всех списков ACL IP, примененных на интерфейсе командой ip access-group.

Пример 2. Стандартный нумерованный список ACL

Для второго примера используется рис. 22.8. Предположим, руководство потребовало фильтровать пакеты, поступающие с серверов, показанных справа, клиентам, показанным слева. Затем руководство приказывает разрешить доступ на сервер S1 для хостов A, B и других хостов в той же подсети, но лишить доступа к этому серверу хосты в подсети хоста C. Впоследствии выдвигается требование лишить доступа на сервер S2 хосты в подсети хоста A, но разрешить его хостам в подсети хоста C. Все фильтруемые пакеты передаются только справа налево, а затем следует требование поместить список ACL на входящий интерфейс F0/0 маршрутизатора R2.

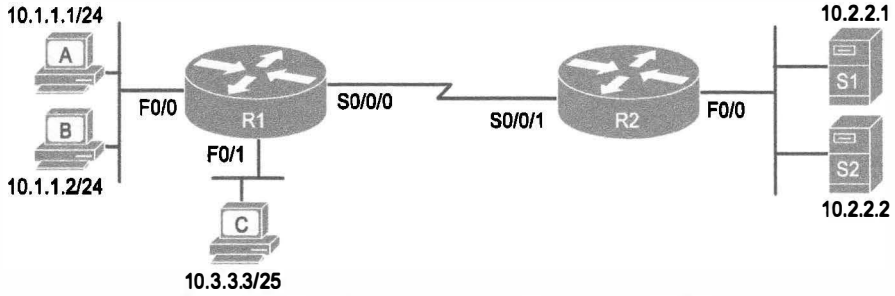


Рис. 22.8. Стандартный нумерованный список ACL. Пример 2

Резюмируя все требования руководства, их можно сократить до следующего списка.

1. Применить список ACL на входящем интерфейсе F0/0 маршрутизатора R2.
2. Разрешить передачу пакетов с сервера S1 на хосты в подсети хоста A.
3. Запретить передачу пакетов с сервера S1 на хосты в подсети хоста C.
4. Разрешить передачу пакетов с сервера S2 на хосты в подсети хоста C.
5. Запретить передачу пакетов с сервера S2 на хосты в подсети хоста A.
6. (Поскольку ничего не было сказано о том, что делать по умолчанию; подразумевается запрет передачи всех остальных пакетов.)

Со стандартным списком ACL требования руководства выполнить не удастся. Рассмотрим, например, очевидную команду для требования номер 2: `access-list 2 permit 10.2.2.1`. Она разрешает передачу всего трафика для отправителя с IP-адресом 10.2.2.1 (сервера S1). Следующее требование — запретить передачу пакетов с того же IP-адреса! Даже при добавлении другой команды, которая проверяла бы IP-адрес отправителя 10.2.2.1, маршрутизатор никогда не дошел бы до нее, поскольку при поиске в списке он использует логику первого соответствия. Нельзя проверять одновременно IP-адреса и получателя, и отправителя, так как стандартные списки управления доступом не могут проверять IP-адрес получателя.

Для решения этой задачи нужно сменить руководство! Если серьезно, то следует заново обдумать задачу и изменить требования. В реальной жизни, вероятно, был бы применен расширенный список ACL, как обсуждается в главе 23, который позволяет проверять как IP-адрес отправителя, так и получателя.

Но для обсуждения стандартных списков управления доступом предположим, что руководство позволило изменить требования. Для начала используем два выходных списка доступа, оба на маршрутизаторе R1. Каждый разрешает передачу трафика от сервера, которому позволено общаться с подсетью, локальной для каждого интерфейса, со следующими измененными требованиями.

1. Использовать исходящий список ACL на интерфейсе F0/0 маршрутизатора R1, разрешающий передачу пакетов с сервера S1 и запрещающий передачу всех остальных пакетов.
2. Использовать исходящий список ACL на интерфейсе F0/1 маршрутизатора R1, разрешающий передачу пакетов с сервера S2 и запрещающий передачу всех остальных пакетов.

Удовлетворяющая этим требованиям конфигурация приведена в примере 22.3.

Пример 22.3. Альтернативная конфигурация на маршрутизаторе R1

```
access-list 2 remark This ACL permits server S1 traffic to host A's
subnet
access-list 2 permit 10.2.2.1
!
access-list 3 remark This ACL permits server S2 traffic to host C's
subnet
access-list 3 permit 10.2.2.2
!
interface F0/0
 ip access-group 2 out
!
interface F0/1
  ip access-group 3 out
```

Как выделено в примере, решение со списком ACL номер 2 разрешает весь трафик от сервера S1, а применяется он для пакетов, выходящих на интерфейс F0/0 маршрутизатора R1. Весь остальной трафик отвергается, поскольку конец списка ACL подразумевает запрет передачи. Кроме того, список ACL 3 разрешает передачу трафика от сервера S2, которому затем разрешается покинуть интерфейс F0/1 маршрутизатора R1. Обратите также внимание на то, что решение демонстрирует применение параметра `access-list remark`, позволяющего оставлять в списке ACL текстовую документацию.

ВНИМАНИЕ!

Когда маршрутизаторы применяют списки ACL для фильтрации пакетов в исходящем направлении, как демонстрирует пример 22.3, маршрутизатор проверяет пакеты, которые перенаправляет через этот список ACL. Но он не фильтрует пакеты, создаваемые самим маршрутизатором. К таким пакетам относятся сообщения протокола маршрутизации OSPF, а также пакеты, посланные командами `ping` и `traceroute` на данном маршрутизаторе.

Советы по проверке, поиску и устранению неисправностей

Поиск и устранение неисправностей в списках ACL IPv4 требуют большого внимания. В частности, следует быть готовым по адресу и шаблону маски уверенно предсказать соответствующие им адреса. В этом помогут практические задания, приведенные далее в этой главе. Кроме того, еще несколько советов могут оказаться полезными в проверке и выявлении проблем ACL на экзаменах.

Сначала, используя пару инструментальных средств, можно выяснить, распознает ли маршрутизатор пакеты или нет. В примере 22.2 уже было показано, что IOS хранит статистику о пакетах, соответствующих каждой строке списка ACL. Кроме того, если в конец команды `access-list` добавить ключевое слово `log`, операционная система IOS начнет выдавать регистрационные сообщения с некой статистикой о соответствиях данной конкретной строке ACL. И статистика, и регистрационные сообщения могут быть полезны при выяснении, какой строке в ACL соответствует пакет.

Пример 22.4 демонстрирует модифицированную версию списка ACL 2 из примера 22.3, но на сей раз с добавленным ключевым словом `log`. В конце примера представлено типичное регистрационное сообщение, демонстрирующее результат распознавания пакета на основании IP-адреса отправителя 10.2.2.1 и адреса получателя 10.1.1.1 (как указано в ACL).

Пример 22.4. Создание регистрационных сообщений для статистики ACL

```
R1# show running-config
! Строки пропущены для краткости
access-list 2 remark This ACL permits server S1 traffic to host A's
subnet
access-list 2 permit 10.2.2.1 log
!
interface F0/0
ip access-group 2 out
R1#


Feb 4 18:30:24.082: %SEC-6-IPACCESSLOGNP: list 2 permitted 0 10.2.2.1 ->
10.1.1.1, 1 packet
```

Каждый раз, исследуя ACL, прежде чем перейти к подробностям логики распознавания, уделите время обдумыванию обоих интерфейсов, на которых установлены списки ACL, и направлению передачи пакета. Иногда логика распознавания безупречна, но список ACL установлен на неправильном интерфейсе или в неправильном направлении.

На рис. 22.9, например, показан тот же список ACL, что и на рис. 22.7. Первая строка этого списка ACL соответствует конкретному адресу хоста 10.1.1.1. Если этот список ACL установлен на маршрутизаторе R2 как входящий, то его интерфейс S0/0/1 вполне может работать, поскольку посланные хостом 10.1.1.1 пакеты (см. слева на рисунке) поступают на его интерфейс S0/0/1. Но если маршрутизатор R2 установит список ACL 1 на интерфейс F0/0 для входящих пакетов, то он никогда не распознает пакет с IP-адресом отправителя 10.1.1.1, поскольку посланные хостом 10.1.1.1 пакеты никогда не будут поступать на этот интерфейс. Благодаря топологии сети посланные хостом 10.1.1.1 пакеты будут покидать маршрутизатор R2 через интерфейс F0/0, но никогда не будут поступать на него.

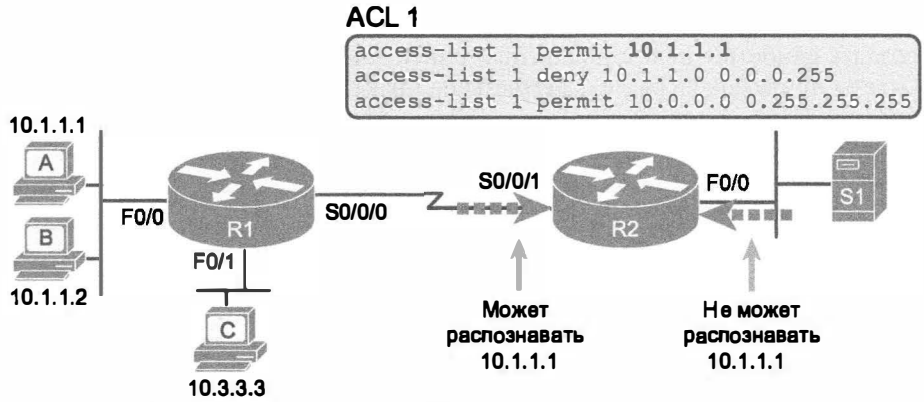


Рис. 22.9. Пример проверки интерфейса и направления при установке ACL

Практические задания на применение стандартных списков ACL

Некоторые темы экзамена CCENT и CCNA, такие как создание подсетей, требуют больше практических навыков, чем другие. Выполнение практических зада-

ний со списками ACL также будет полезно, поскольку они требуют вычисления диапазонов номеров, что, в свою очередь, требует использования математического механизма и знания некоторых процессов.

В данном разделе представлено несколько практических задач и советов с двух точек зрения. Во-первых, этот раздел требует построения однострочных стандартных списков управления доступом, распознающих некоторые пакеты. Во-вторых, он требует интерпретации существующих команд ACL, чтобы описать, каким пакетам будет соответствовать список. Оба навыка пригодятся на экзамене.

Практические задания на создание команд списка управления доступом

Задания этого раздела рассчитаны на знание синтаксиса команд `access-list`, особенно выбора правильной логики распознавания. Эти знания будут полезны при изучении расширенных и именованных списков ACL в следующей главе.

Ниже дано несколько важнейших советов, которые стоит учитывать при выборе параметров соответствия для любой команды `access-list`.

Ключевая тема

Советы по выбору параметров соответствия для любой команды `access-list`

- Чтобы распознать определенный адрес, следует указать только его.
- Чтобы распознать любой адрес или все адреса, используйте ключевое слово `any`.
- Чтобы распознавать только один, два или три первых октета адреса, используйте маски WC 0.255.255.255, 0.0.255.255 и 0.0.0.255 соответственно. Кроме того, удостоверьтесь, что у параметра адреса отправителя в октетах шаблона установлены нули.
- Чтобы распознать подсеть, используйте в качестве параметра адреса отправителя идентификатор подсети и найдите маску WC путем вычитания маски подсети в формате DDN из 255.255.255.255.

Требования для нескольких практических заданий приведены в табл. 22.1. Задание: создать однострочный стандартный список ACL, который соответствует пакетам. Ответы приведены в разделе "Ответы на практические задания главы".

Таблица 22.1. Построение однострочных стандартных списков управления доступом. Задания

Задание	Требования
1	Пакеты от хоста 172.16.5.4
2	Пакеты от хостов со значением 192.168.6 в первых трех октетах
3	Пакеты от хостов со значением 192.168 в первых двух октетах
4	Пакеты от любого хоста
5	Пакеты из подсети 10.1.200.0/21
6	Пакеты из подсети 10.1.200.0/27
7	Пакеты из подсети 172.20.112.0/23
8	Пакеты из подсети 172.20.112.0/26
9	Пакеты из подсети 192.168.9.64/28
10	Пакеты из подсети 192.168.9.64/30

Обратное проектирование: от списков ACL к диапазону адресов

Некоторые экзаменационные вопросы могут потребовать не выбирать оператор ACL, а интерпретировать существующие команды `access-list`. Для ответа на вопросы этого типа необходимо выяснить диапазон IP-адресов, соответствующий определенной комбинации адреса и шаблона маски, в каждом операторе ACL.

Ключевая тема

Как вычислить диапазон чисел, используемых в параметрах адреса отправителя и шаблона маски

При некоторых допущениях, уместных для экзамена CCENT и CCNA, вычисление диапазона адресов, соответствующих команде ACL, может быть относительно простым. Нижняя граница диапазона — поле адреса, а верхнюю границу диапазона находят за счет добавления адреса к маске WC. Вот и все.

Например, в команде `access-list 1 permit 172.16.200.0 0.0.7.255` значение нижней границы диапазона — 172.16.200.0 — взято непосредственно из команды. Для вычисления верхней границы диапазона достаточно добавить это число к маске WC следующим образом:

```
172.16.200.0
+ 0. 0.  7.255
--------------
172.16.207.255
```

Теперь рассмотрим существующие команды `access-list` в табл. 22.2. Для каждого случая запишите конкретный IP-адрес или диапазон IP-адресов, соответствующий команде.

Таблица 22.2. Поиск IP-адресов и диапазонов, соответствующих существующим спискам ACL

Задание	Команда, для которой нужно найти диапазон адресов отправителя
1	`access-list 1 permit 10.7.6.5`
2	`access-list 2 permit 192.168.4.0 0.0.0.127`
3	`access-list 3 permit 192.168.6.0 0.0.0.31`
4	`access-list 4 permit 172.30.96.0 0.0.3.255`
5	`access-list 5 permit 172.30.96.0 0.0.0.63`
6	`access-list 6 permit 10.1.192.0 0.0.0.31`
7	`access-list 7 permit 10.1.192.0 0.0.1.255`
8	`access-list 8 permit 10.1.192.0 0.0.63.255`

Интересно то, что операционная система IOS позволяет пользователю CLI вводить команду `access-list` в режиме конфигурации и (потенциально) изменить параметр адреса прежде, чем поместить команду в файл `running-config` конфигурации. Этот процесс только поиска диапазона соответствующих команде `access-list` адресов ожидает, что команда `access-list` исходит от маршрутизатора, так что любые изменения будут внесены.

Изменение, которое операционная система IOS может внести благодаря команде `access-list`, предполагает преобразование в нуль любых октетов адреса, для ко-

торых октет шаблона маски равен 255. Например, при шаблоне маски 0.0.255.255 операционная система IOS игнорирует последние два октета. Она ожидает, что поле адреса закончится двумя нулями. В противном случае операционная система IOS примет команду `access-list`, но изменит последние два октета адреса на нуль. Как демонстрирует пример 22.5, выводится адрес 10.1.1.1, но шаблоном маски является 0.0.255.255.

Пример 22.5. Операционная система IOS изменяет поле адреса в команде `access-list`

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# access-list 21 permit 10.1.1.1 0.0.255.255
R2(config)# ^Z
R2#
R2# show ip access-lists
Standard IP access list 21
    10 permit 10.1.0.0, wildcard bits 0.0.255.255
```

Математический механизм поиска диапазона адресов полагается на тот факт, что либо команда полностью правильна, либо операционная система IOS уже исправила эти октеты адреса на нуль, как показано в примере.

ВНИМАНИЕ!

Наиболее популярные маски WC не чередуют двоичные 0 и 1. В этой книге подразумевается использование только таких типов масок WC. Однако маски WC, чередующие 0 и 1, вполне возможны, но они нарушают простой метод вычисления диапазона адресов. По мере повышения уровня знаний (до CCIE) будьте готовы изучить тему списков ACL существенно глубже.

Обзор

Резюме

- Списки управления доступом (ACL) IPv4 позволяют сетевым инженерам идентифицировать пакеты различных типов.
- Списки ACL IPv4 выполняют множество функций в маршрутизаторах Cisco, но чаще всего они используются как фильтр пакетов.
- Маршрутизаторы Cisco способны применить логику ACL к пакетам в том месте, где пакеты IP поступают на интерфейс, или в том, где они покидают интерфейс. Другими словами, списки ACL ассоциируются с интерфейсом и направлением потока передачи пакетов (входящим или исходящим).
- Каждый список ACL IP состоит из одной или нескольких команд конфигурации, каждая из которых содержит список значений, искомых в заголовках пакета. Как правило, команда ACL использует следующую логику: "искать указанные значения в заголовке пакета, а если они найдены, отказаться от пакета". (Действием может быть разрешение на прохождение пакета вместо запрета.) Конкретно список ACL задает поиск полей заголовка, которые должны быть уже знакомы, включая IP-адреса отправителя и получателя, а также номера портов TCP и UDP.
- При использовании списков ACL IP для фильтрации пакетов может быть выбрано только одно из двух действий. Команды конфигурации используют ключевые слова `deny` (запретить) и `permit` (разрешить), означающие (соответственно) отказ от пакета или разрешение его передачи, как будто список ACL не существует.
- Списки ACL используют логику первого соответствия. Как только обнаруживается соответствие пакета одной из строк списка ACL, маршрутизатор предпринимает действие, указанное в этой строке списка, и прекращает дальнейшее сравнение.
- Стандартные нумерованные списки ACL IP используют следующую глобальную команду:

```
access-list {1-99 | 1300-1999} {permit | deny} параметры_
соответствия
```

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Barney — это хост с IP-адресом 10.1.1.1 в подсети 10.1.1.0/24. Какие из следующих действий могут осуществляться в результате настройки конфигурации стандартного списка управления доступом IP? (Выберите два ответа.)

   А) Проверка конкретного IP-адреса отправителя.

   Б) Проверка IP-адресов от 10.1.1.1 до 10.1.1.4 с помощью одной команды `access-list`, без проверки других IP-адресов.

В) Проверка всех IP-адресов в подсети хоста Barney с помощью одной команды `access-list`, без проверки других IP-адресов.

Г) Проверка только IP-адреса получателя пакета.

2. В каком из следующих ответов приведено допустимое число для стандартной нумерации в списках ACL IP? (Выберите два ответа.)

   А) 1987.

   Б) 2187.

   В) 187.

   Г) 87.

3. Какой из следующих шаблонов масок наиболее удобен для проверки всех пакетов IP в подсети 10.1.128.0 с маской 255.255.255.0?

   А) 0.0.0.0.

   Б) 0.0.0.31.

   В) 0.0.0.240.

   Г) 0.0.0.255.

   Д) 0.0.15.0.

   Е) 0.0.248.255.

4. Какой из следующих шаблонов масок наиболее удобен для проверки всех пакетов IP в подсети 10.1.128.0 с маской 255.255.240.0?

   А) 0.0.0.0.

   Б) 0.0.0.31.

   В) 0.0.0.240.

   Г) 0.0.0.255.

   Д) 0.0.15.255.

   Е) 0.0.248.255.

5. У списка ACL 1 есть три оператора в следующем порядке со значениями адреса и шаблона маски: 1.0.0.0 0.255.255.255, 1.1.0.0 0.0.255.255 и 1.1.1.0 0.0.0.255. Если маршрутизатор попытается распознать пакет, полученный с IP-адреса 1.1.1.1, используя этот список ACL, то какой оператор ACL маршрутизатор посчитает соответствующим пакету?

   А) Первый.

   Б) Второй.

   В) Третий.

   Г) Будет запрещен по завершении списка ACL

6. Какая из следующих команд `access-list` соответствует всем пакетам, посланных хостами из подсети 172.16.5.0/25?

   А) `access-list 1 permit 172.16.0.5 0.0.255.0.`

   Б) `access-list 1 permit 172.16.4.0 0.0.1.255.`

   В) `access-list 1 permit 172.16.5.0.`

   Г) `access-list 1 permit 172.16.5.0 0.0.0.128.`

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 22.3.

Таблица 22.3. Ключевые темы главы 22

Элемент	Описание	Страница
Параграф	Общие правила расположения и направления для списков ACL	641
Рис. 22.3	Четыре основные категории списков ACL IPv4 операционной системы Cisco IOS	643
Параграф	Логика первого соответствия используется всеми списками ACL	644
Список	Логика шаблона маски для десятичных чисел 0 и 255	646
Список	Логика шаблона маски при поиске соответствия подсети	647
Список	Этапы планирования и реализации стандартных списков ACL IP	648
Список	Советы по созданию логики соответствия для поля адреса отправителя в команде списка управления доступом	649
Список	Этапы реализации стандартного списка ACL IP	650
Список	Советы по выбору параметров соответствия для любой команды access-list	656
Параграф	Как вычислить диапазон чисел, используемых в параметрах адреса отправителя и шаблона маски	657

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

стандартный список доступа (standard access list), шаблон маски (wildcard mask)

Практические задания в приложении K

Дополнительные практические задания и ответы на них приведены в приложении K. Это приложение можете найти на веб-странице книги в формате PDF.

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 22.4 приведен список команд конфигурации, а в табл. 22.5 пользовательские команды главы. Фактически команды стоит запомнить, чтобы лучше понять содержимое главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 22.4. Команды конфигурации главы 22

Команда	Описание
`access-list` *номер_списка_управления_доступом* `{deny | permit}` *отправитель* [*шаблон_маски_отправителя*] [log]	Глобальная команда для стандартного нумерованного списка управления доступом. Используются номера 1–99 или 1300–1999 включительно
`access-list` *номер_списка_управления_доступом* `remark` *текст*	Задает комментарий, помогающий запомнить, для чего предназначен список управления доступом
`ip access-group` *номер* `{in | out}`	Подкоманда интерфейса, применяющая списки управления доступом

Таблица 22.5. Пользовательские команды главы 22

Команда	Описание
`show ip interface` [*тип номер*]	Выводит информацию о списках управления доступом, примененных на интерфейсе
`show access-lists` [*номер_списка_управления_доступом* | *имя_списка*]	Показывает сведения о введенных в конфигурацию списках управления доступом для всех протоколов
`show ip access-list` [*номер_списка_управления_доступом* | *имя_списка*]	Показывает сведения о списках управления доступом IP

Практика

Ответы на практические задания главы

Табл. 22.6 содержит ответы на задачи, перечисленные ранее в табл. 22.1.

Таблица 22.6. Построение однострочных стандартных списков управления доступом. Ответы

Задача	Ответы
1	`access-list 1 permit 172.16.5.4`
2	`access-list 2 permit 192.168.6.0 0.0.0.255`
3	`access-list 3 permit 192.168.0.0 0.0.255.255`
4	`access-list 4 permit any`
5	`access-list 5 permit 10.1.200.0 0.0.7.255`
6	`access-list 6 permit 10.1.200.0 0.0.0.31`
7	`access-list 7 permit 172.20.112.0 0.0.1.255`
8	`access-list 8 permit 172.20.112.0 0.0.0.63`
9	`access-list 9 permit 192.168.9.64 0.0.0.15`
10	`access-list 10 permit 192.168.9.64 0.0.0.3`

Табл. 22.7 содержит ответы на задания, приведенные в табл. 22.2.

Таблица 22.7. Диапазоны адресов для задач в табл. 22.2. Ответы

Задача	Диапазон адресов
1	Один адрес: 10.7.6.5
2	192.168.4.0 – 192.168.4.127
3	192.168.6.0 – 192.168.6.31
4	172.30.96.0 – 172.30.99.255
5	172.30.96.0 – 172.30.96.63
6	10.1.192.0 – 10.1.192.31
7	10.1.192.0 – 10.1.193.255
8	10.1.192.0 – 10.1.255.255

Ответы на контрольные вопросы:

1 А и В. **2** А и Г. **3** Г. **4** Д. **5** А. **6** Б.

Глава 23

Расширенные списки управления доступом и защита устройств

Маршрутизаторы Cisco используют *списки управления доступом* (Access Control List — ACL) IPv4 для многих целей: распознавания пакетов для принятия решения о фильтрации, распознавания пакетов для *трансляции сетевых адресов* (Network Address Translation — NAT), распознавания пакетов для принятия решения о *качестве обслуживания* (Quality of Service — QoS) и др.

Большинство списков ACL IP являются либо стандартными, либо расширенными. Стандартные списки управления доступом ищут соответствие только по IP-адресу отправителя, а расширенные — по множеству полей заголовка пакета. В то же время списки ACL IP могут быть нумерованными или именованными. На рис. 23.1 в продолжение темы предыдущей главы представлены категории и основные возможности каждого из типов.

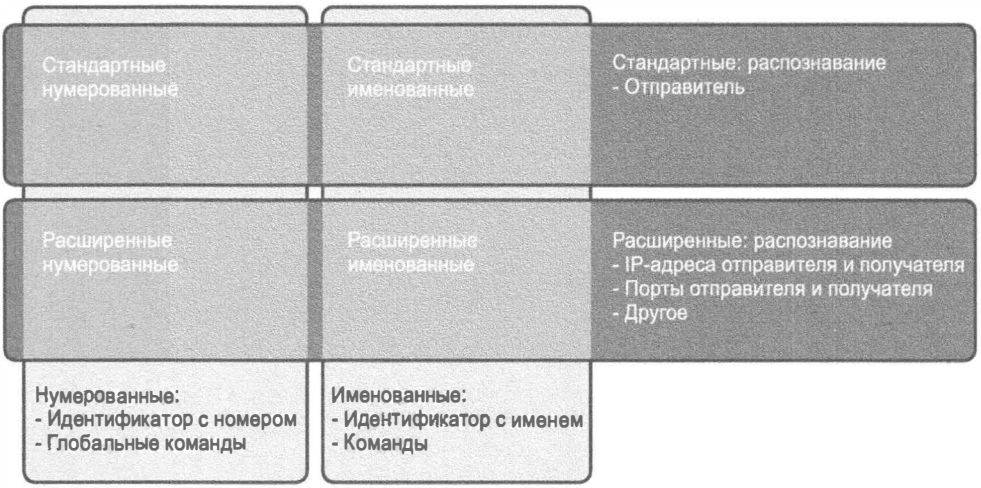


Рис. 23.1. Сравнения типов списков ACL IP

В этой главе обсуждаются три других категории списков ACL, кроме стандартных нумерованных списков ACL IP, а также несколько других тем, связанных с обеспечением защиты маршрутизаторов и коммутаторов Cisco.

В этой главе рассматриваются следующие экзаменационные темы

Службы IP

- Типы, средства и приложения ACL.
  - Порядковые номера.
  - Редактирование.
  - Расширенные.
  - Именованные.
  - Нумерованные.
- Настройка и проверка ACL в сетевой среде.
  - Именованные.
  - Нумерованные.
- Настройка и проверка NTP на клиенте.

Защита сетевых устройств

- Настройка и проверка средств защиты сетевых устройств.
  - Транспорт.
  - Отключение telnet.
  - Физическая защита.
  - Установка собственной сети VLAN, отличной от VLAN 1.
  - Настройка и проверка списков ACL для фильтрации сетевого трафика.
- Настройка и проверка списков ACL для ограничения обращений по протоколам telnet и SSH к маршрутизатору.

Поиск и устранение неисправностей

- Поиск неисправностей и решение проблем списков ACL.
  - Статистика.
  - Разрешенные сети.
  - Направление.
  - Интерфейс.

Основные темы

Расширенные нумерованные списки управления доступом IP

Расширенные списки управления доступом IP очень похожи на стандартные нумерованные списки ACL, обсуждавшиеся в предыдущей главе. Как и в случае стандартных списков, расширенные списки управления доступом применяются для пакетов, либо входящих на интерфейс, либо исходящих из интерфейса. Система IOS проводит поиск в этом списке последовательно. Расширенные списки доступа также используют логику первого соответствия, поскольку маршрутизатор останавливает поиск по списку, как только обнаруживается первый соответствующий оператор, и предпринимает определенное в нем действие. Все эти особенности верны также для стандартных нумерованных списков управления доступом (и именованных списков ACL).

Расширенные списки доступа отличаются от стандартных большим разнообразием полей заголовка пакета, применяемых для распознавания. Один оператор расширенного списка ACL может задать проверку нескольких элементов заголовка пакета, требуя точного соответствия всех параметров правилам данного оператора ACL. Такая мощная логика распознавания делает расширенные списки управления доступом и более полезными, и более сложными, чем стандартные списки ACL.

Распознавание протокола, IP-адреса отправителя и получателя

Подобно стандартному нумерованному списку ACL IP, расширенный нумерованный список ACL IP также использует глобальную команду `access-list`. Синтаксис тот же, по крайней мере, в использовании ключевых слов `permit` и `deny`. Список параметров распознавания команд, конечно, отличается. В частности, расширенная команда ACL `access-list` требует трех параметров соответствия: тип протокола IP, IP-адрес отправителя и IP-адрес получателя.

Поле Protocol заголовка IP идентифицирует заголовок, который следует за заголовком IP. На рис. 23.2 представлены расположение поля Protocol в заголовке IP, концепция указания на тип следующего заголовка, а также некоторые подробности заголовка IP для справки.

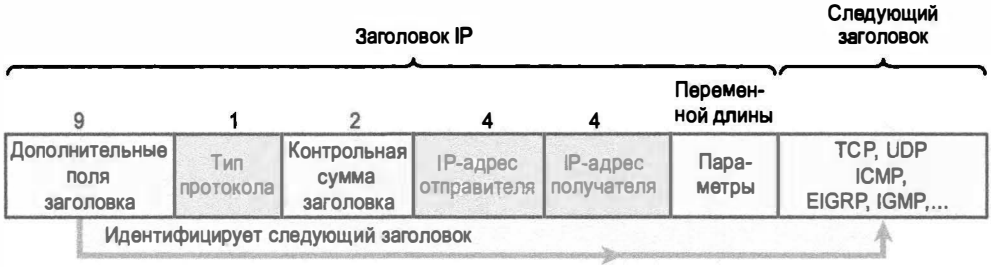


Рис. 23.2. Заголовок IP с выделенными полями, необходимыми для расширенного списка ACL

Операционная система IOS требует настройки параметров для трех частей, выделенных на рис. 23.2. Для типа протокола используется такое ключевое слово, как `tcp`, `udp` или `icmp`, для пакетов IP, у которых после заголовка IP есть заголовок

TCP, UDP или ICMP соответственно. Либо можно использовать ключевое слово `ip`, означающее "все пакеты IP". Необходимо также настроить несколько значений для расположенных далее полей IP-адреса отправителя и получателя. Для этих полей используется тот же синтаксис и параметры распознавания IP-адреса, которые обсуждались в главе 22. Синтаксис представлен на рис. 23.3.

Ключевая тема

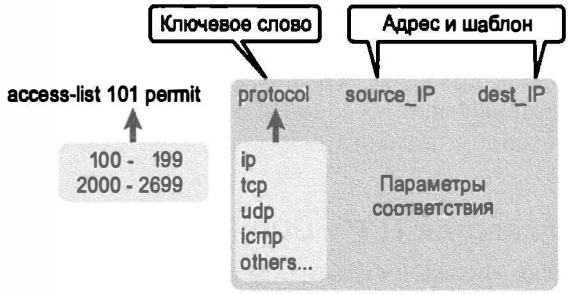


Рис. 23.3. Синтаксис расширенных списков ACL с необходимыми полями

ВНИМАНИЕ!

В распознавании полей IP-адресов отправителя и получателя есть одно различие со стандартными списками ACL: при поиске определенного IP-адреса расширенные списки ACL требуют использования ключевого слова `host`. Нельзя просто указать один IP-адрес.

В табл. 23.1 приведено несколько команд `access-list`, которые используют только необходимые параметры распознавания. Можно закрыть правую сторону таблицы и использовать ее для обучения или изучить объяснения, чтобы понять логику некоторых типичных команд.

Таблица 23.1. Команды `access-list` расширенного списка управления доступом и пояснения к применяемым принципам распознавания

Оператор `access-list`	Какой пакет соответствует правилу
`access-list 101 deny tcp any any`	Любой пакет IP, имеющий заголовок TCP
`access-list 101 deny udp any any`	Любой пакет IP, имеющий заголовок UDP
`access-list 101 deny icmp any any`	Любой пакет IP, имеющий заголовок ICMP
`access-list 101 deny ip host 1.1.1.1 host 2.2.2.2`	Все пакеты IP от хоста 1.1.1.1, следующие на хост 2.2.2.2, независимо от заголовка после заголовка IP
`access-list 101 deny udp 1.1.1.0 0.0.0.255 any`	Все пакеты IP, у которых есть заголовок UDP после заголовка IP, следующие из подсети 1.1.1.0/24 к любому получателю

Последняя запись в табл. 23.2 позволяет сделать важное заключение о том, как операционная система IOS обрабатывает расширенные списки ACL.

Ключевая тема

Важнейшая особенность логики расширенных списков ACL

В команде `access-list` *расширенного списка ACL, чтобы пакет считался соответствующим команде, все параметры распознавания должны соответствовать полям пакета.*

Например, в последней строке табл. 23.1 команда проверяет наличие заголовка UDP, IP-адреса отправителя из подсети 1.1.1.0/24 и любого IP-адреса получателя. Если бы был исследован пакет с IP-адресом отправителя 1.1.1.1, то он прошел бы проверку IP-адреса отправителя, но если бы он имел заголовок TCP, а не UDP, то не соответствовал бы данной команде `access-list`. Соответствовать должны все параметры.

Проверка номеров портов TCP и UDP

Расширенные списки управления доступом позволяют также исследовать части заголовков TCP и UDP, в частности, поля номера порта получателя и отправителя. Номера портов идентифицируют приложение, которое посылает или получает данные.

Чаще всего проверяют порты, являющиеся стандартными портами, используемыми серверами. Например, веб-серверы используют по умолчанию стандартный порт 80. На рис. 23.4 представлено расположение номеров портов в заголовке TCP после заголовка IP.

Ключевая тема

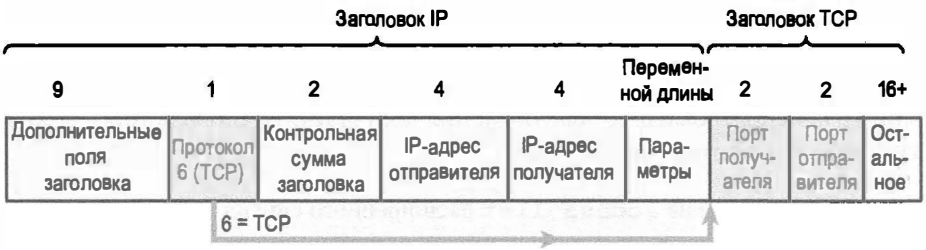


Рис. 23.4. Поля номеров портов в заголовке TCP, сопровождающем заголовок IP

Когда команда расширенного списка ACL включает ключевое слово `tcp` или `udp`, она может (необязательно) проверить порт отправителя и/или получателя. Для этого синтаксис использует для номеров портов ключевые слова равно (equal), не равно (not equal), меньше (less), больше (greater) и диапазон (range). Кроме того, команда может использовать литеральные или десятичные номера портов либо более удобные ключевые слова для некоторых общеизвестных портов приложений. Позиции полей портов отправителя и получателя в команде `access-list`, а также ключевых слов для номеров портов представлены на рис. 23.5.

В качестве примера рассмотрим простую сеть, показанную на рис. 23.6. Сервер FTP находится справа на рисунке, а клиент — слева. На рис. 23.6 показаны синтаксические конструкции списка управления доступом, применяемые для проверки соответствия перечисленных ниже типов пакетов критериям списка.

- Пакеты, которые включают заголовок TCP.
- Пакеты, отправленные из подсети клиента.

- Пакеты, отправленные в подсеть сервера.
- Пакеты с портом получателя TCP, равным 21 (порт управления сервера FTP).

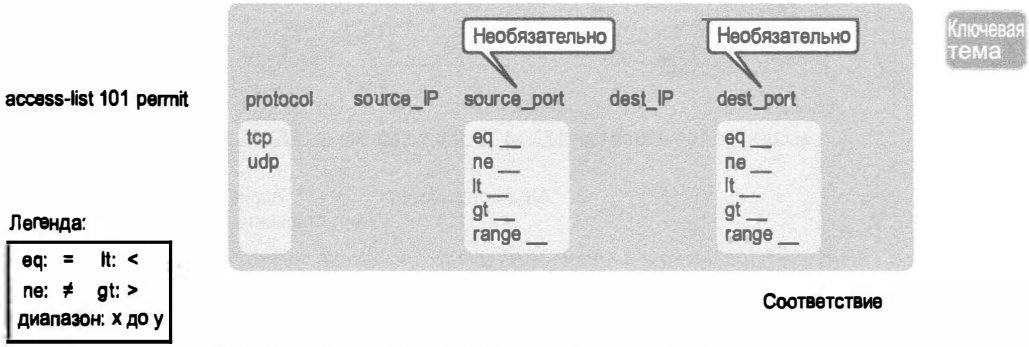


Рис. 23.5. Расширенный синтаксис команд ACL с номерами портов при использовании протокола TCP или UDP

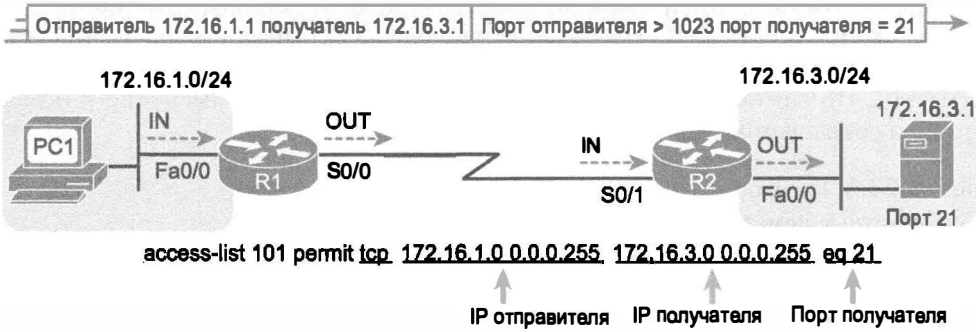


Рис. 23.6. Фильтрация пакетов на основе номера порта получателя

Чтобы полностью разобраться в том, как происходит проверка номера порта получателя на основе параметра `eq 21`, рассмотрим пакеты, движущиеся слева направо, от компьютера PC1 к серверу. Предположим, на сервере используется зарезервированный порт 21 (порт управления протокола FTP), поэтому у пакета, отправленного компьютером PC1, в своем заголовке TCP указан порт получателя 21. Синтаксическая конструкция команды в списке управления доступом включает параметр `eq 21` *после* IP-адреса получателя. Позиция после параметров адреса получателя важна: она свидетельствует о том, что параметр `eq 21` следует сравнивать с портом получателя пакета. В результате оператор списка управления доступом, показанный на рис. 23.6, будет успешно сопоставлен с пакетом (в том числе номер порта получателя 21), если он задан в любом из четырех вариантов, обозначенных четырьмя стрелками на этом рисунке.

С другой стороны, на рис. 23.7 показан обратный поток, в котором происходит передача ответного пакета от сервера компьютеру PC1. В этом случае в заголовке TCP пакета указан порт отправителя 21, поэтому в списке управления доступом необходимо предусмотреть проверку значения номера порта отправителя, равного 21, а сам список управления доступом должен быть задан для других интерфейсов.

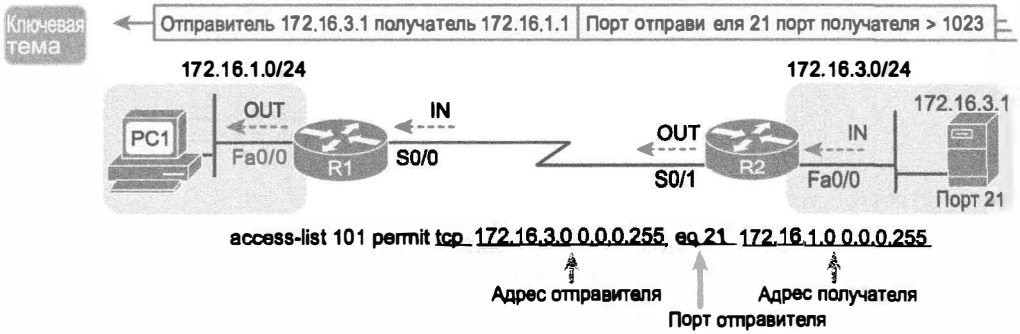


Рис. 23.7. Фильтрация пакетов по данным о номере порта отправителя

В экзаменационных вопросах по спискам ACL и распознаванию номеров портов сначала рассмотрите позицию параметра и направление, в котором будет применен список ACL. Направление определяет, передают ли пакет на сервер или с сервера. Теперь можно решить, требуется ли проверить в пакете порт отправителя или получателя, с учетом того, что проверяемая служба использует стандартный порт.

Для справки в табл. 23.2 перечислены наиболее известные номера портов, а также соответствующие им приложения и протоколы транспортного уровня. Следует учитывать, что синтаксис команд `access-list` допускает применение и номеров портов, и сокращенных вариантов имен приложений.

Таблица 23.2. Распространенные приложения и соответствующие им стандартные номера портов

Номер порта	Протокол	Приложение	Ключевое слово в команде `access-list`
20	TCP	FTP	`data ftp-data`
21	TCP	Управление сервером FTP	`ftp`
22	TCP	SSH	—
23	TCP	Telnet	`telnet`
25	TCP	SMTP	`Smtp`
53	UDP, TCP	DNS	`Domain`
67, 68	UDP	DHCP	`nameserver`
69	UDP	TFTP	`Tftp`
80	TCP	HTTP (WWW)	`www`
110	TCP	POP3	`pop3`
161	UDP	SNMP	`Snmp`
443	TCP	SSL	—
16 384–32 767	UDP	Передача голоса (VoIP) и видео на основе RTP	—

В табл. 23.3 приведено несколько примеров команд `access-list` с распознаванием на основании номера порта. Закройте правую сторону таблицы и попытайтесь охарактеризовать пакеты, соответствующие каждой команде. Затем проверьте свои ответы по правой стороне таблицы.

Таблица 23.3. Примеры команд расширенных списков ACL и объяснение логики

Оператор access-list	Чему соответствует
`access-list 101 deny tcp any gt 1023 host 10.1.1.1 eq 23`	Пакеты с заголовком TCP, любым IP-адресом отправителя, с номером порта отправителя больше (`gt`) 1023, IP-адресом получателя 10.1.1.1 и номером порта получателя, равным (`eq`) 23
`access-list 101 deny tcp any host 10.1.1.1 eq 23`	То же, что и выше, но подходят любые порты отправителя, поскольку этот параметр в данном случае пропущен
`access-list 101 deny tcp any host 10.1.1.1 eq telnet`	То же, что и выше, но вместо указания порта 23 используется ключевое слово `telnet`
`access-list 101 deny udp 1.0.0.0 0.255.255.255 lt 1023 any`	Пакет с отправителем в сети 1.0.0.0/8, использующий протокол UDP с портом отправителя, номер которого меньше (`lt`) 1023, и любым IP-адресом получателя

Настройка расширенных списков управления доступом

Как уже было сказано, расширенные списки управления доступом позволяют проводить проверку многих полей из различных заголовков в пакете IP, поэтому синтаксис соответствующей команды вряд ли можно легко подытожить в виде одной универсальной команды.

Однако для подготовки к экзамену CCNA можно ориентироваться на две синтаксические конструкции, приведенные в табл. 23.4.

Таблица 23.4. Команды настройки конфигурации расширенных списков управления доступом IP

Команда	Режим настройки конфигурации и описание
`access-list` *номер* {deny | permit} *протокол адрес-отправителя шаблон_маски-отправителя адрес-получателя шаблон_маски-получателя* [log | log-input]	Глобальная команда для расширенных нумерованных списков управления доступом. Используются номера 100–199 или 2000–2699 включительно
`access-list` *номер* {deny | permit} {tcp | udp} *адрес-отправителя шаблон_маски-отправителя* [*оператор* [*порт*]] *адрес-получателя шаблон_маски-получателя* [*оператор* [*порт*]] [established] [log]	Версия команды `access-list` с параметрами, специфическими для протокола TCP или UDP

Процесс настройки конфигурации расширенных списков управления доступом в основных чертах совпадает с аналогичным процессом, используемым для стандартных списков управления доступом. В первую очередь необходимо выбрать положение и направление, чтобы можно было планировать применение параметров списка управления доступом с учетом информации в пакетах, проходящих в выбранном направлении. Настройка конфигурации списка управления доступом осуществляется с помощью команд `access-list`, а по завершении, чтобы задейство-

вать список ACL, используется такая же команда `ip access-group`, применяемая для стандартных списков управления доступом. Все эти этапы отражают то, что происходит со стандартными списками управления доступом; однако при настройке помните о следующих различиях.

Ключевая тема **Советы и рекомендации по проверке портов TCP и UDP с использованием списков ACL**

- Располагайте расширенные списки управления доступом как можно ближе к отправителю пакетов, подлежащих фильтрации. Применение фильтрации ближе к источнику экономит полосу пропускания.
- Помните, что пакет считается соответствующим оператору `access-list`, только при полном совпадении всех параметров в одной из команд `access-list` с соответствующими полями пакета.
- Для расширенной команды `access-list` могут использоваться номера 100–199 или 2000–2699, причем ни один номер не рассматривается как более предпочтительный по отношению к другому.

Расширенные списки управления доступом (пример 1)

Назначение данного примера состоит в более глубоком изучении основного синтаксиса списков. В данном случае предполагается, что для компьютера Bob запрещен доступ ко всем серверам FTP в сети Ethernet маршрутизатора R1 и для компьютера Larry запрещен доступ к веб-серверу Server1. На рис. 23.8 еще раз показана топология сети, которая рассматривалась в предыдущем примере. В примере 23.1 показана конфигурация маршрутизатора R1.

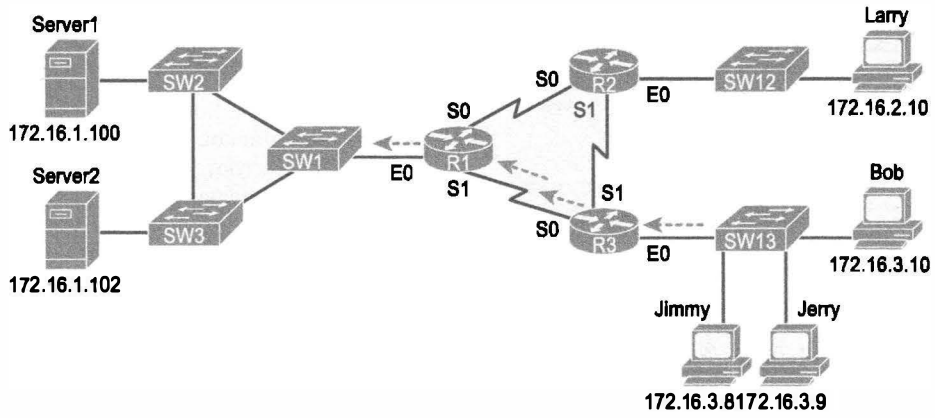


Рис. 23.8. Схема сети для расширенного списка управления доступом (пример 1)

Пример 23.1. Расширенный список управления доступом маршрутизатора R1 (пример 1)

```
interface Serial0
   ip address 172.16.12.1 255.255.255.0
   ip access-group 101 in
```

```
!
interface Serial1
   ip address 172.16.13.1 255.255.255.0
   ip access-group 101 in
!
access-list 101 remark Stop Bob to FTP servers, and Larry to Server1 web
access-list 101 deny tcp host 172.16.3.10 172.16.1.0 0.0.0.255 eq ftp
access-list 101 deny tcp host 172.16.2.10 host 172.16.1.100 eq www
access-list 101 permit ip any any
```

Первый оператор списка управления доступом предотвращает для компьютера Bob доступ к серверам FTP в подсети 172.16.1.0. Второй оператор предотвращает для компьютера Larry доступ к веб-службам на компьютере Server1. Последний оператор разрешает весь прочий трафик.

Прежде всего рассмотрим синтаксис этих операторов, поскольку необходимо описать в нем несколько новых особенностей. Вначале напомним, что номера, применяемые в расширенных списках управления доступом, должны находиться в диапазоне 100–199 или 2000–2699. Вслед за указанием действия ключами `permit` или `deny` должен находиться параметр с обозначением протокола, который указывает, должна ли выполняться проверка всех пакетов IP или только пакетов с заголовками TCP или UDP. Если проверке подлежат номера портов TCP или UDP, то должен быть указан протокол TCP или UDP. (И FTP, и веб используют протокол TCP.)

В этом примере используется ключевое слово `eq`, означающее "равно", для проверки номеров портов получателей, относящихся к протоколу управления FTP (ключевое слово `ftp`), и трафика HTTP (ключевое слово `www`). Безусловно, можно использовать числовые значения, но для наиболее распространенных значений номеров портов более удобной является текстовая версия параметра. (В частности, в случае применения в команде оператора `eq 80` в конфигурации отображается `eq www`.)

Этот пример задействует список ACL на маршрутизаторе R1 в двух местах: на каждом входящем последовательном интерфейсе. Эти расположения удовлетворяют задаче ACL. Как будет описано в конце данной главы, корпорацией Cisco даны некоторые конкретные рекомендации в отношении расположения списков управления доступом. Поэтому в примере 23.2 достигается та же цель, что и в примере 23.1, т.е. предотвращается доступ для компьютера Bob к серверам FTP, находящимся на основном хосте, причем для решения этой задачи применяется список управления доступом, введенный в конфигурацию маршрутизатора R3.

Пример 23.2. Применение в маршрутизаторе R3 расширенного списка управления доступом для блокирования пакетов от компьютера Bob на сервер FTP, находящийся рядом с маршрутизатором R1

```
interface Ethernet0
   ip address 172.16.3.1 255.255.255.0
   ip access-group 103 in
access-list 103 remark deny Bob to FTP servers in subnet 172.16.1.0/24
access-list 103 deny tcp host 172.16.3.10 172.16.1.0 0.0.0.255 eq ftp
access-list 103 permit ip any any
```

Новая конфигурация на маршрутизаторе R3 соответствует задаче фильтрации трафика компьютера Bob, а также общей задаче по расположению списков ACL

ближе к отправителю пакетов. Список ACL 103 на маршрутизаторе R3 очень похож на список ACL 101 маршрутизатора R1 из примера 23.1, но на сей раз он не проверяет критерии, соответствующие трафику компьютера Larry, поскольку он никогда не будет попадать на интерфейс Ethernet 0 маршрутизатора R3. Список ACL 103 фильтрует трафик FTP компьютера Bob в направлении к подсети 172.16.1.0/24 со всем другим трафиком, вводящим на интерфейс E0 маршрутизатора R3 и попадающим затем в общую сеть.

Расширенные списки управления доступом (пример 2)

В примере 23.3, в основе которого лежит сеть, показанная на рис. 23.9, демонстрируется еще один способ использования расширенных списков управления доступом IP. В данном примере используются следующие критерии.

- Компьютеру Sam запрещен доступ к подсети компьютеров Bugs или Daffy.
- Для хостов в сети Ethernet маршрутизатора Seville запрещен доступ к хостам сети Ethernet маршрутизатора Yosemite.
- Обмен данными между всеми прочими хостами разрешен.

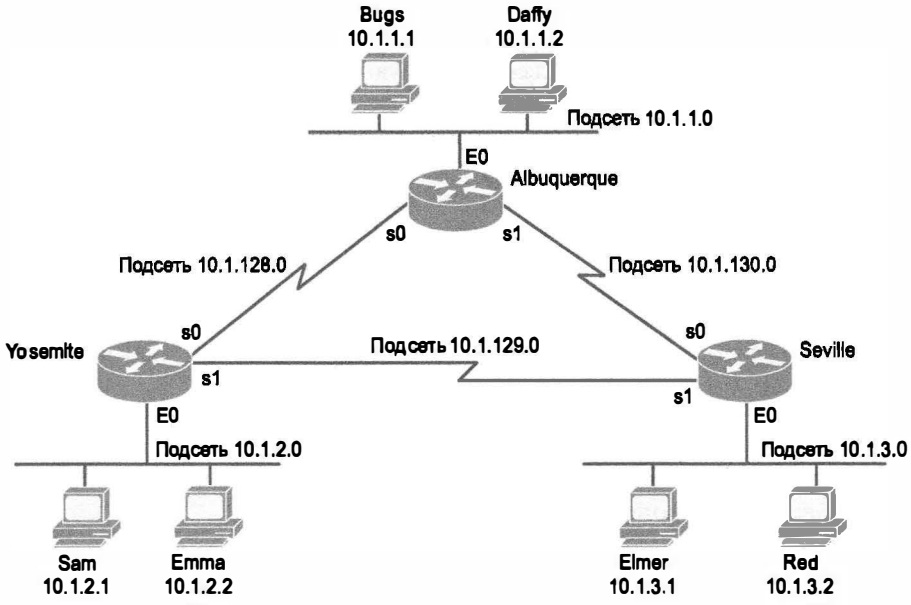


Рис. 23.9. Схема сети для расширенного списка управления доступом (пример 2)

Пример 23.3. Конфигурация маршрутизатора Yosemite для расширенного списка управления доступом (пример 2)

```
interface ethernet 0
   ip access-group 110 in
!
access-list 110 deny ip host 10.1.2.1 10.1.1.0 0.0.0.255
access-list 110 deny ip 10.1.2.0 0.0.0.255 10.1.3.0 0.0.0.255
access-list 110 permit ip any any
```

Эта конфигурация позволяет решить поставленную задачу с помощью всего лишь нескольких операторов и вместе с тем соответствует рекомендациям по проектированию сетей от корпорации Cisco, поскольку предусматривает размещение расширенных списков управления доступом как можно ближе к отправителям трафика. Список управления доступом фильтрует пакеты, поступающие на интерфейс E0 маршрутизатора Yosemite, который является первым интерфейсом маршрутизатора, куда поступают пакеты, отправленные с компьютера Sam. Если маршрут между маршрутизатором Yosemite и другими подсетями со временем изменится, список ACL все еще будет применим. Кроме того, требование по фильтрации, которое указано в качестве второго пункта задачи (согласно которому необходимо предотвратить для хостов локальной сети Seville доступ к маршрутизатору Yosemite), выполняется с помощью второго оператора `access-list`. Предотвращение возможности передачи пакетов из подсети локальной сети маршрутизатора Yosemite в подсеть локальной сети маршрутизатора Seville по существу исключает эффективный обмен данными между этими двумя подсетями. Еще один вариант состоит в том, что в конфигурации маршрутизатора Seville могут быть реализованы противоположные требования.

Практические задания на создание команд списков управления доступом

В табл. 23.5 приведены практические задания, способные помочь приобрести навыки в синтаксисе команды `access-list` расширенных списков ACL, особенно в выборе правильной логики распознавания. Задание заключается в создании однострочного расширенного списка ACL, который соответствует пакетам. Ответы находятся в разделе "Ответы на практические задания главы". Обратите внимание на то, что если критерий упоминает определенный протокол прикладной программы, например "веб-клиент", то это означает соответствие именно этому протоколу.

Таблица 23.5. Создание однострочных расширенных списков ACL. Задания

Задание	Критерий
1	От веб-клиента 10.1.1.1 к веб-серверу в подсети 10.1.2.0/24
2	От клиента Telnet 172.16.4.3/25 к серверу Telnet в подсети 172.16.3.0/25. Соответствие также всем хостам в подсети клиента
3	Сообщения ICMP из подсети 192.168.7.200/26 всем хостам в подсети 192.168.7.14/29
4	От веб-сервера в подсети 10.2.3.4/23/23 к клиентам в подсети 10.4.5.6/22
5	От подсети сервера Telnet 172.20.1.0/24 к клиентам в подсети 172.20.44.1/23
6	Пакеты от веб-клиента подсети 192.168.99.99/28, следующие на веб-сервер в подсети 192.168.176.0/28. Соответствие также всем хостам в подсети клиента
7	Сообщения ICMP из подсети 10.55.66.77/25 всем хостам в подсети 10.66.55.44/26
8	Любой и каждый пакет IPv4

Именованные списки ACL и их редактирование

На данный момент читатель должен полностью разобраться в основных понятиях списков управления доступом IP, применяемых в операционной системе IOS.

В настоящем разделе рассматривается ряд усовершенствований списков управления доступом в системе IOS: именованные списки управления доступом и их редактирование с помощью порядковых номеров. Безусловно, обе указанные возможности являются важными и нужными, но не вносят какие-либо дополнительные функции по отношению к тем, с помощью которых маршрутизатор может фильтровать трафик. Вместо этого именованные списки управления доступом и порядковые номера списков управления доступом позволяют проще запоминать их имена и редактировать существующие списки управления доступом, если потребуется их изменить.

Именованные списки управления доступом

Именованные и нумерованные списки ACL IP имеют много сходств. Они применяются не только для фильтрации пакетов, но и для многих других целей. Точно так же, подобно стандартным и расширенным нумерованным спискам ACL, которые отличаются возможностями распознавания пакетов, именованные списки ACL могут быть стандартными и расширенными.

Первоначально именованные списки ACL имели три существенных отличия от нумерованных списков ACL.

Ключевая тема

Различия между именованными и нумерованными списками ACL

- Вместо номеров для идентификации списков ACL используются имена, облегчающие запоминание причин их применения.
- Для определения действий и параметров распознавания используются команды подсистемы ACL, а не глобальные команды.
- Средства редактирования ACL, позволяющие пользователю CLI удалять отдельные строки из списка ACL и вставлять новые.

Изучить конфигурацию именованных списков ACL довольно легко, достаточно преобразовать нумерованный список ACL в именованный эквивалент. Такое преобразование простого стандартного списка ACL номер 1 из трех строк приведено на рис. 23.10. Чтобы создать три подкоманды `permit` для именованного списка ACL, достаточно скопировать части трех команд нумерованного списка ACL, начиная с ключевого слова `permit`.

Рис. 23.10. Конфигурации нумерованного и именованного списков ACL

Единственная действительно новая часть конфигурации именованных списков ACL — это глобальная команда конфигурации `ip access-list`, которая определяет, является ли список ACL стандартным или расширенным, а также задает имя. Она также переводит пользователя в режим конфигурации ACL, как видно в приме-

ре 23.4. В режиме конфигурации ACL настраиваются команды `permit`, `deny` и `remark`, которые отражают синтаксис команд `access-list` нумерованных списков ACL. У стандартных именованных списков ACL эти команды соответствуют синтаксису стандартных нумерованных списков ACL, а у расширенных именованных списков ACL — синтаксису команд расширенных нумерованных списков ACL.

Пример 23.4 демонстрирует конфигурацию именованного списка ACL, а затем удаление из него одной строки. Обратите особое внимание на приглашения, которые демонстрируют переход в режим конфигурации ACL.

Пример 23.4. Конфигурация именованного списка доступа

```
Router# configure terminal
Enter configuration commands, one per line. End with Ctrl-Z.
Router(config)# ip access-list extended barney
Router(config-ext-nacl)# permit tcp host 10.1.1.2 eq www any
Router(config-ext-nacl)# deny udp host 10.1.1.1 10.1.2.0 0.0.0.255
Router(config-ext-nacl)# deny ip 10.1.3.0 0.0.0.255 10.1.2.0 0.0.0.255
Router(config-ext-nacl)# deny ip 10.1.2.0 0.0.0.255 10.2.3.0 0.0.0.255
Router(config-ext-nacl)# permit ip any any
Router(config-ext-nacl)# interface serial1
Router(config-if)# ip access-group barney out
Router(config-if)# ^Z
Router# show running-config
Building configuration...

Current configuration:


! строки пропущены для краткости

interface serial 1
  ip access-group barney out
!
ip access-list extended barney
permit tcp host 10.1.1.2 eq www any
deny   udp host 10.1.1.1 10.1.2.0 0.0.0.255
deny   ip 10.1.3.0 0.0.0.255 10.1.2.0 0.0.0.255
deny   ip 10.1.2.0 0.0.0.255 10.2.3.0 0.0.0.255
permit ip any any
```

Пример 23.4 начинается с создания списка управления доступом, обозначенного именем “barney”. С помощью команды `ip access-list extended barney` создается список управления доступом, которому присваивается имя “barney”, после чего пользователь переходит в режим настройки конфигурации списка управления доступом. Эта команда содержит также сведения для системы IOS о том, что “barney” представляет собой расширенный список управления доступом. Затем следуют пять операторов с ключевыми словами `permit` и `deny`, которые определяют критерии проверки пакетов и указывают, какие действия должны быть выполнены в случае совпадения параметров пакета с правилом списка. Команда `show running-config` выводит данные о конфигурации, определяемой именованным списком управления доступом, перед тем как удаляется отдельная запись из списка.

Именованные списки ACL позволяют пользователю удалять и добавлять новые строки из режима конфигурации ACL. Пример 23.5 демонстрирует, как команда `no deny ip ...` удаляет одну запись из таблицы ACL. Обратите внимание, что вывод команды `show access-list` в конце примера выводит содержимое списка ACL с четырьмя правилами `permit` и `deny` вместо пяти.

Пример 23.5. Удаление одной команды из именованного списка ACL

```
Router# configure terminal
Enter configuration commands, one per line. End with Ctrl-Z.
Router(config)# ip access-list extended barney
Router(config-ext-nacl)# no deny ip 10.1.2.0 0.0.0.255 10.2.3.0 0.0.0.255
Router(config-ext-nacl)# ^Z
Router# show access-list

Extended IP access list barney
    10 permit tcp host 10.1.1.2 eq www any
    20 deny   udp host 10.1.1.1 10.1.2.0 0.0.0.255
    30 deny   ip 10.1.3.0 0.0.0.255 10.1.2.0 0.0.0.255
    50 permit ip any any
```

Редактирование списков управления доступом с использованием порядковых номеров

Применение нумерованных списков управления доступом было предусмотрено в системе IOS уже в самых первых моделях маршрутизаторов Cisco. Однако на протяжении многих лет и многих версий операционной системы IOS возможности редактирования нумерованных списков ACL IP оставались слабыми. Например, чтобы просто удалить строку из списка ACL, пользователь должен был удалить весь список ACL, а затем снова создать его.

Современные версии операционной системы IOS позволяют пользователям CLI легко редактировать и нумерованные, и именованные списки ACL. Сначала компания Cisco предоставила улучшенные средства редактирования только для именованных списков ACL, а впоследствии предоставила их и для нумерованных. Данный раздел посвящен этим улучшенным средствам редактирования ACL, которыми операционная система IOS обладала с версии 12.3. На момент написания этой книги она уже была относительно устаревшей.

Средства редактирования ACL используют порядковые номера, добавляемые к каждому оператору `permit` или `deny` списка ACL, а также номера, представляющие последовательности операторов в списке ACL. Порядковые номера ACL предоставляют следующие средства и для нумерованных, и для именованных списков ACL.

Ключевая тема

Средства, предоставляемые операционной системой IOS 12.3 порядковыми номерами ACL

- **Новый стиль конфигурации для нумерованных списков.** Нумерованные списки ACL используют теперь стиль конфигурации, как у именованных, а также традиционный стиль; для расширенного редактирования списков ACL необходим новый стиль.

- **Удаление отдельных строк**. С помощью команды `no` *порядковый_номер* можно удалять в списке управления доступом отдельные строки операторов `permit` или `deny`.
- **Вставка новых строк**. Добавляемые команды `permit` и `deny` можно задавать в конфигурации с указанием порядкового номера, определяя местонахождение оператора в списке управления доступом.
- **Автоматическая нумерация**. Операционная система IOS сама добавляет командам порядковые номера при настройке, даже если они неизвестны инженеру.

Чтобы иметь возможность удалять и добавлять строки и в нумерованных, и в именованных списках управления доступом, необходимо использовать один общий стиль конфигурирования и вводить такие же команды, которые служат для работы с именованными списками управления доступом. Единственное различие в синтаксисе заключается в том, используется ли для обозначения списка имя или номер. В примере 23.6 показана конфигурация стандартного нумерованного списка управления доступом IP, демонстрирующая указанный альтернативный стиль настройки конфигурации. На основании этого примера можно судить, насколько широкие возможности для редактирования предоставляют порядковые номера списка управления доступом. В данном примере выполняются описанные ниже действия.

Этап 1 Создание нумерованного списка ACL 24, состоящего из трех команд `permit`, с использованием конфигурации в новом стиле

Этап 2 Вывод с помощью команды `show ip access-list` трех команд `permit` с порядковыми номерами 10, 20 и 30

Этап 3 Удаление инженером только второй команды `permit` с использованием подкоманды `no 20` для списка управления доступом, в которой указан порядковый номер 20

Этап 4 Проверка с помощью команды `show ip access-list` того, что список управления доступом содержит теперь только две строки (с порядковыми номерами 10 и 30)

Этап 5 Добавление инженером новой команды `permit` к началу списка управления доступом с использованием команды конфигурирования списка управления доступом `5 deny 10.1.1.1`

Этап 6 Повторная проверка с помощью команды `show ip access-list` правильности внесенных изменений, которая показывает, что на сей раз имеются три команды `permit` с порядковыми номерами 5, 10 и 30

ВНИМАНИЕ!

В отношении данного примера следует отметить, что пользователь не выходит из режима конфигурации устройства и вместо этого использует команду `do` для передачи системе IOS указания, что команда режима EXEC `show ip access-list` должна быть выполнена без выхода из режима настройки конфигурации.

Пример 23.6. Редактирование списков управления доступом с использованием порядковых номеров

```
! Этап 1. В конфигурацию введен стандартный нумерованный список
! управления доступом IP, состоящий из трех строк.
R1# configure terminal
```

```
Enter configuration commands, one per line. End with Ctrl-Z.
R1(config)# ip access-list standard 24
R1(config-std-nacl)# permit 10.1.1.0 0.0.0.255
R1(config-std-nacl)# permit 10.1.2.0 0.0.0.255
R1(config-std-nacl)# permit 10.1.3.0 0.0.0.255

! Этап 2. Отображение содержимого списка управления доступом без выхода
! из режима настройки конфигурации.
R1(config-std-nacl)# do show ip access-list 24
Standard IP access list 24
    10 permit 10.1.1.0, wildcard bits 0.0.0.255
    20 permit 10.1.2.0, wildcard bits 0.0.0.255
    30 permit 10.1.3.0, wildcard bits 0.0.0.255

! Этап 3. Удаление строки с порядковым номером 20 в условиях дальнейшего
! пребывания в режиме настройки конфигурации списка ACL 24.
R1(config-std-nacl)# no 20

! Этап 4. Повторное отображение содержимого списка управления доступом
! без выхода из режима настройки конфигурации.
! Обратите внимание на то, что теперь строка с номером 20 отсутствует в
! результатах вывода.
R1(config-std-nacl)# do show ip access-list 24
Standard IP access list 24
    10 permit 10.1.1.0, wildcard bits 0.0.0.255
    30 permit 10.1.3.0, wildcard bits 0.0.0.255

! Этап 5. Вставка новой первой строки в список управления доступом.
R1(config-std-nacl)# 5 deny 10.1.1.1

! Этап 6. Отображение содержимого списка управления доступом в последний
! раз, что позволяет видеть новый оператор (с порядковым
! номером 5), находящуюся на первом месте в списке.
R1(config-std-nacl)# do show ip access-list 24
Standard IP access list 24
    5  deny   10.1.1.1
    10 permit 10.1.1.0, wildcard bits 0.0.0.255
    30 permit 10.1.3.0, wildcard bits 0.0.0.255
```

Несмотря на то что в примере 23.6 используется нумерованный список ACL, редактирование (добавление и удаление) записей именованного списка ACL осуществляется точно так же.

Конфигурация нумерованных или именованных списков ACL

Резюмируя тему нумерованных списков ACL, следует заметить, что более новые версии операционной системы IOS фактически допускают два способа настройки нумерованных списков ACL: традиционный, подразумевающий использование глобальных команд `access-list` и продемонстрированный ранее в примерах 23.1–23.3, а также способ, аналогичный используемому для именованных списков ACL, продемонстрированный в примере 23.6.

Как ни странно, но операционная система IOS всегда хранит нумерованные списки ACL в первоначальном стиле конфигурации, как глобальные команды `access-list`, независимо от используемого метода настройки списка ACL. Пример 23.7 демонстрирует эти факты, продолжая пример 23.6 следующими дополнительными этапами.

Этап 7 Вывод инженером результатов настройки конфигурации (с помощью команды `show running-config`), в которых отображаются команды конфигурации в старом стиле, даже несмотря на то, что сам список управления доступом был создан с помощью команд в новом стиле

Этап 8 Добавление инженером нового оператора в конце списка управления доступом с использованием глобальной команды конфигурации `access-list 24 permit 10.1.4.0 0.0.0.255` в старом стиле

Этап 9 Подтверждение с помощью команды `show ip access-list` того, что команда `access-list` в старом стиле, выполненная на предыдущем этапе, добавлена в соответствии с правилом, согласно которому она должна появиться только в конце списка управления доступом

Этап 10 Отображение инженером конфигурации для подтверждения того, что все части списка ACL, конфигурация которых была настроена и с помощью команд в новом стиле, и с помощью команд в старом стиле, присутствуют в выводе одного и того же списка управления доступом в старом стиле (с помощью команды `show running-config`)

Пример 23.7. Добавление к нумерованному списку управления доступом новых команд конфигурации и его отображение

```
! Этап 7. Фрагмент конфигурации, относящийся к списку ACL 24.
R1# show running-config
! Единственными отображаемыми строками являются строки из списка ACL 24
access-list 24 deny   10.1.1.1
access-list 24 permit 10.1.1.0 0.0.0.255
access-list 24 permit 10.1.3.0 0.0.0.255

! Этап 8. Добавление новой глобальной команды access-list 24
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# access-list 24 permit 10.1.4.0 0.0.0.255
R1(config)# ^Z

! Этап 9. Повторное отображение содержимого списка управления доступом с
! порядковыми номерами. Обратите внимание на то, что даже
! новому оператору был автоматически присвоен порядковый номер.
R1# show ip access-list 24
Standard IP access list 24
    5  deny   10.1.1.1
    10 permit 10.1.1.0, wildcard bits 0.0.0.255
    30 permit 10.1.3.0, wildcard bits 0.0.0.255
    40 permit 10.1.4.0, wildcard bits 0.0.0.255
!
! Этап 10. Конфигурация нумерованного списка управления доступом
! по-прежнему остается определенной с помощью команд конфигурации в
! старом стиле.
R1# show running-config
! Единственными отображаемыми строками являются строки из списка ACL 24
access-list 24 deny   10.1.1.1
access-list 24 permit 10.1.1.0 0.0.0.255
access-list 24 permit 10.1.3.0 0.0.0.255
access-list 24 permit 10.1.4.0 0.0.0.255
```

Защита маршрутизатора и коммутатора

В данном разделе рассматривается множество небольших тем, относящихся к защите маршрутизаторов и коммутаторов. Некоторые из средств подразумевают использование для лучшей защиты маршрутизатора или коммутатора списков ACL, а другие полагаются лишь на рекомендации Cisco по защите устройств.

В следующем разделе обсуждается применение пароля, отключение ненужных служб и Telnet, защита доступа по VTY (Telnet и SSH) при помощи списков ACL, выбор наилучших расположений для списков ACL и *протокол синхронизации сетевого времени* (Network Time Protocol — NTP).

Защита паролем интерфейса CLI

В главах 7 и 8 обсуждалась защита доступа к CLI маршрутизатора и коммутатора с использованием различных паролей. В качестве напоминания на рис. 23.11 показано, где операционная система IOS может потребовать ввода пароля при входе в CLI или переходе в другой режим: на консоли, vty и при переходе из пользовательского режима в привилегированный.

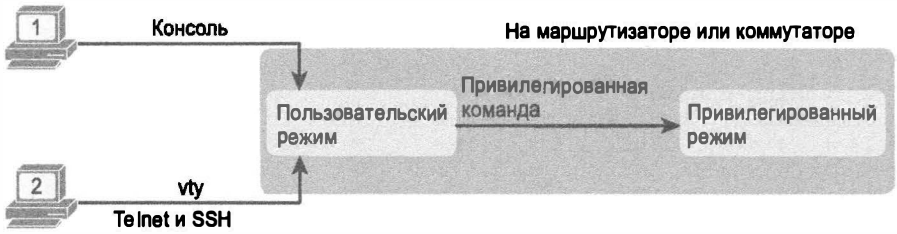


Рис. 23.11. Ситуация, в которой необходимо применять рефлексивные списки управления доступом

Некоторые из рекомендаций по защите интерфейса CLI маршрутизатора и коммутатора приведены ниже. В главе 8 уже объяснялось значение всех этих команд с примерами применения некоторых из них. В списке ниже указаны также причины использования различных паролей.

- Используйте команду `enable secret` вместо комбинации команд `enable password` и `service password-encryption`. Будучи отображенными командой `show running-config`, оба результата выглядят одинаково. Однако команда `enable secret` обеспечивает более мощное шифрование пароля, в то время как пароли, зашифрованные командой `service password-encryption`, могут быть легко взломаны.
- Избегайте использования для консоли и VTY простого пароля, задаваемого командой `login`, поскольку этот метод не идентифицирует отдельного пользователя.
- Идеально использовать для аутентификации пользователя CLI такой внешний сервер аутентификации, как RADIUS. Но в случае необходимости используйте локально введенные команды `username secret`, скрывающие пароли за хеш-кодом (как это делает команда `enable secret`).

- Отключите поддержку входящих соединений Telnet, так как они передают пароли в виде открытого текста, позволяя злоумышленнику перехватить пакеты и узнать пароль. Вместо этого настройте маршрутизатор и коммутатор так, чтобы позволить только соединения SSH при помощи команды `transport input ssh` в режиме линии VTY.

Хотя все эти действия тоже важны, все же необходимо обеспечить надежную физическую защиту сетевых устройств. Физически они должны находиться в помещении, доступ к которому имеет только уполномоченный персонал. Как только злоумышленник получит физический доступ к маршрутизатору или коммутатору, он сможет отключать кабели и питание устройства и даже удалить пароли с консоли, что позволит ему впоследствии получать доступ к устройству дистанционно.

Отключение служб

У операционной системы Cisco IOS, как и у любой другой OS, должны быть некие стандартные настройки. Изначально включено множество служб, которые операционная система IOS поддерживает по тем или иным причинам. Хотя большинство этих служб поддерживается из лучших побуждений, хакеры могут использовать их для взлома сети или получения информации, которая пригодится при следующей атаке. Поэтому для снижения риска имеет смысл создать план защиты для каждой используемой в сети OS, найти все стандартные настройки, способные создать брешь в защите, и изменить их.

Компания Cisco дает несколько рекомендаций о том, что включать, а что отключать на операционной системе IOS (и для маршрутизатора, и для коммутатора). В данном разделе описано несколько таких элементов, чтобы дать лишь общее представление о них, поскольку это тема из сертификационных экзаменов Cisco по защите устройств.

ВНИМАНИЕ!

Более подробные рекомендации по защите маршрутизаторов и коммутаторов можно найти на сайте Cisco.com, если ввести в поле поиска строку "Guide to Harden Cisco IOS Devices".

Операционная система Cisco IOS предоставляет графический интерфейс пользователя (GUI), способный выполнить ту же работу, что и интерфейс CLI. Для этого операционная система IOS работает как веб-сервер. Изначально операционная система IOS разрешает выполнение веб-службы HTTP, которая не шифрует данные (HTTP), но есть возможность настроить службу HTTP, которая действительно шифрует данные (HTTPS). Рекомендация? Отключить службу HTTP и разрешить только службу HTTPS, если действительно предполагается позволить пользователям подключаться к маршрутизатору или коммутатору, используя веб-браузер.

Обсуждавшийся в главе 10 протокол обнаружения устройств Cisco (CDP) позволяет устройствам на том же канале связи изучать основную информацию друг о друге. Но эта базовая информация способна помочь злоумышленнику узнать полезную информацию о сети. Поэтому компания Cisco рекомендует отключить протокол CDP на всех интерфейсах, соединенных с небезопасными частями сети. Для большей безопасности протокол CDP можно отключить глобально.

ВНИМАНИЕ!

В реальных сетях будьте осторожны, выключая протокол CDP на коммутаторах LAN! Удостоверьтесь, что все устройства соединяются с коммутатором без проблем. Многие телефоны IP требуют применения протокола CDP в соединении между телефоном и коммутатором, а без него телефон не работает.

У операционной системы IOS есть ряд служб, которые она относит к малым (small) службам. Например, служба Echo — это одна из таких служб. Она действует как утилита `ping`, использующая эхо-запросы и эхо-ответы ICMP, но, в отличие от сообщений ICMP, приложение Echo использует протоколы TCP или UDP. Считайте ее утилитой `ping`, проверяющей также и транспортный уровень (чего настоящая утилита `ping` не делает). Операционная система IOS должна выполнять службу Echo, чтобы быть готовой отвечать на эхо-запросы этих типов.

У некоторых версий IOS эти службы изначально выключены, а у некоторых нет. Для полной уверенности отключите малые службы как TCP, так и UDP.

Пример 23.8 демонстрирует отключение на маршрутизаторе R1 функций, упомянутых в этом разделе. Комментарии в примере объясняют отдельные команды конфигурации.

Пример 23.8. Отключение служб IOS

```
! Отключить службу HTTP
R1(config)# no ip http server

! Отключить малые службы TCP и UDP
R1(config)# no service tcp-small-servers
R1(config)# no service udp-small-servers

! Отключить протокол CDP только на одном интерфейсе;
! Команда no cdp run отключит протокол CDP глобально
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# no cdp enable
R1(config-if)# ^Z
```

Управление доступом по протоколам Telnet и SSH с помощью списков ACL

Когда внешний пользователь подключается к маршрутизатору или коммутатору по Telnet или SSH, операционная система IOS использует канал vty для предоставления ему пользовательского соединения. Применив списки ACL к этим входящим соединениям, операционная система IOS может фильтровать адреса хостов IPv4, от которых маршрутизаторы или коммутаторы могут принимать соединения telnet или SSH.

Предположим, например, что весь технический персонал сети использует подсеть 10.1.1.0/24 и устройства только из этой подсети должны быть в состоянии установить сеансы telnet с любым из маршрутизаторов Cisco в сети. В таком случае конфигурация в примере 23.9 применяется на каждом маршрутизаторе, чтобы лишить доступа все IP-адреса не из этой подсети.

Пример 23.9. Управление доступом через соединения vty с использованием команды access-class

```
line vty 0 4
  login
  password cisco
  access-class 3 in
!
! Следующая глобальная команда соответствует пакетам IPv4 с адресом
! отправителя, начинающимся на 10.1.1
  access-list 3 permit 10.1.1.0 0.0.0.255
```

В команде `access-class` применяется ссылка на средства проверки, заданные с помощью команды `access-class 3 in`. Ключевое слово `in` указывает, что речь идет о входящих запросах на создание соединений Telnet и SSH с маршрутизатором, иными словами, о подключении к маршрутизатору с помощью этого протокола удаленного соединения. В том виде, в каком он задан в конфигурации, список ACL 3 проверяет IP-адрес отправителя в пакетах, относящихся к запросам на создание соединений Telnet.

Операционная система IOS позволяет также использовать списки ACL для фильтрации исходящих соединений Telnet и SSH. Рассмотрим, например, пользователя, который использовал протокол telnet или SSH для подключения к CLI и теперь находится в пользовательском или привилегированном режиме. При исходящем фильтре VTY операционная система IOS применит логику ACL, если пользователь попытается применить команду `telnet` или `ssh` для подключения с *локального устройства* к другому устройству.

Чтобы настроить исходящий список ACL VTY, используйте команду `access-class` *`acl`* `out` в режиме конфигурации VTY. После настройки маршрутизатор фильтрует попытки текущих пользователей vty использовать команды `telnet` и `ssh` для инициализации новых соединений с другими устройствами.

Из этих двух возможностей (защита входящих и исходящих соединений) защита входящих соединений безусловно важней и применяется чаще. Но для полноты картины следует заметить, что у исходящих списков ACL VTY есть удивительная особенность. При использовании ключевого слова `out` стандартный список ACL IP, перечисленный в команде `ip access-class`, фактически контролирует *IP-адрес получателя*, а не отправителя. Таким образом, он осуществляет фильтрацию на основании устройства, с которым команда `telnet` или `ssh` пытается установить соединение.

Принципы использования списков управления доступом

Списки ACL могут быть прекрасным инструментом для улучшения защиты сети, но инженерам придется подумать о некоторых более распространенных проблемах, прежде чем просто настраивать списки ACL для исправления проблем. Компания Cisco дает приведенные ниже общие рекомендации для курсов, на которых основаны экзамены CCNA.

Ключевая тема

Рекомендации по реализации списков ACL

- Размещайте расширенные списки управления доступом как можно ближе к отправителю пакета, чтобы сразу же отбрасывать определенные типы пакетов.
- Размещайте стандартные списки управления доступом как можно ближе к получателю пакетов, поскольку они часто уничтожают пакеты, которые не должны быть уничтожены, если находятся ближе к отправителю.
- Размещайте более специфичные (т.е. узкие) правила проверки ближе к началу списка управления доступом.
- Прежде чем вносить изменения в список управления доступом, удалите список управления доступом на том интерфейсе, на котором он задан (с помощью команды `no ip access-group`).

Первый пункт относится к концепции расположения списков ACL. Если цель состоит в фильтрации пакетов, то применение критериев, допускающих прохождение только определенных пакетов ближе к их отправителю, означает, что ненужные пакеты с самого начала не будут расходовать пропускную способность сети, что, по-видимому, должно способствовать повышению производительности сети (и действительно способствует). Поэтому корпорация Cisco рекомендует располагать расширенные списки управления доступом настолько близко к отправителю, насколько это возможно.

Второй пункт, расположение поближе к получателю, казалось бы, противоречит первому пункту, по крайней мере, для стандартных списков ACL. Почему? Дело в том, что стандартные списки управления доступом обеспечивают только проверку IP-адреса отправителя, поэтому, как правило, будучи размещенными ближе к отправителю, они фильтруют больший объем трафика, чем предполагалось. Предположим, например, что компьютеры Fred и Barney разделены четырьмя маршрутизаторами. Если фильтрация трафика компьютера Barney, направляемого на компьютер Fred, будет осуществляться на первом маршрутизаторе, то пакеты от компьютера Barney не смогут достичь каких-либо хостов, связанных со всеми прочими тремя маршрутизаторами. В связи с этим на курсах ICND2 корпорации Cisco применяется общая рекомендация, согласно которой стандартные списки управления доступом должны находиться как можно ближе к получателю, чтобы была предотвращена фильтрация того трафика, для которого не предназначен данный фильтр.

Третий пункт списка, согласно которому более специфичные правила размещаются ближе к началу списка ACL, вероятно, менее подвержен ошибкам. Предположим, например, что в списке ACL сначала расположена команда, разрешающая прохождение всего трафика для подсети 10.1.1.0/24, а затем запрещающая трафик на хост 10.1.1.1. При этом посланные на хост 10.1.1.1 пакеты соответствовали бы первой команде и никогда не распознавались бы второй, более специфической командой. Размещение более специфических команд вначале позволит избежать подобных ошибок.

Наконец, корпорация Cisco рекомендует удалять списки управления доступом на интерфейсах и только после этого вносить изменения в оператор этого списка. К счастью, если какой-либо список управления доступом IP включен в интерфейсе

с помощью команды `ip access-group`, после чего удален весь список управления доступом, система IOS прекращает фильтрацию каких-либо пакетов. (В более ранних версиях IOS так было не всегда!) При этом сразу после добавления команд в список управления доступом начинается фильтрация пакетов системой IOS.

Предположим, например, что разрешено применение списка ACL 101 на интерфейсе S0/0/0 для исходящих пакетов. Затем происходит удаление списка 101, чтобы было разрешено прохождение всех пакетов. После этого вводится одна команда `access-list 101`. Сразу после нажатия клавиши <Enter> создается список, и маршрутизатор начинает фильтровать все пакеты, исходящие из интерфейса S0/0/0, на основе этого однострочного списка. Таким образом, даже если должен был введен в действие длинный список управления доступом, может оказаться, что в течение какого-то времени происходит фильтрация пакетов, которые не должны были быть отфильтрованными! Поэтому наилучший путь — запретить применение списка на интерфейсе, внести в него изменения, а затем снова разрешить его применение на интерфейсе.

Протокол синхронизации сетевого времени

Заключительная тема этой главы — способы решения проблем при помощи регистрационных сообщений, отправляемых маршрутизаторами и коммутаторами. Первоначально решение этой проблемы могло быть связано и не с защитой, но фактически оно играет ключевую роль при поиске и сборе доказательств о произошедшей атаке.

Маршрутизаторы и коммутаторы выдают регистрационные сообщения в ответ на разные события. Например, при отказе интерфейса устройство создает регистрационные сообщения. Стандартно операционная система IOS посылает эти сообщения на консольный порт, но можно перенастроить ее на обработку регистрационных сообщений множеством других способов. Некоторые из них см. в главе 19.

Одной из возможностей обработки регистрационных сообщений является использование службы сервера системного журнала, когда маршрутизаторы и коммутаторы перенаправляют копии всех регистрационных сообщений на сервер системного журнала, который сохраняет их. При централизованном хранении, если возникнут проблемы, персонал поддержки сети всегда сможет просмотреть сообщения от всех устройств и выяснить их причину. Концепция сервера системного журнала приведена на рис. 23.12.

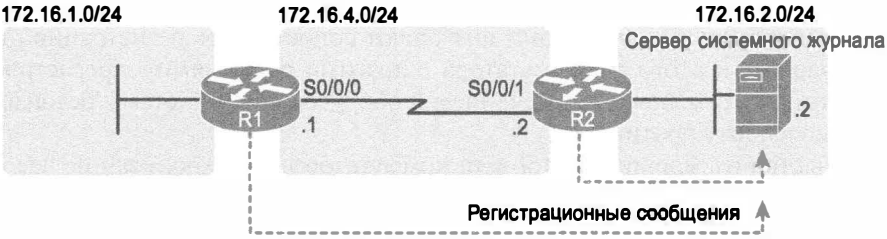


Рис. 23.12. Пример сети с сервером системного журнала

Теперь, владея этой базовой информацией о сообщениях системного журнала, подумайте о времени. В частности, о времени суток. У каждого устройства есть сис-

темные часы, и большинство регистрационных сообщений указывают дату и время в сообщениях. Почему? Чтобы сетевой инженер, когда будет впоследствии просматривать эти сообщения, точно знал, когда какое событие произошло.

Протокол синхронизации сетевого времени (Network Time Protocol — NTP) позволяет любому устройству, такому как маршрутизатор или коммутатор, синхронизировать свои часы. Если часы всех сетевых устройств синхронизированы, то сообщения могут быть отсортированы и просмотрены по дате и времени, что ускоряет и упрощает поиск причин произошедшего.

Чтобы убедиться в важности синхронизации часов, рассмотрим пример 23.10, где маршрутизаторы R1 и R2 не синхронизируют свои часы. Предположим, проблема в последовательном канале связи между двумя этими маршрутизаторами. Сетевой инженер просматривает все регистрационные сообщения на обоих устройствах, как показано на рис. 23.12. Но когда он видит сообщения на маршрутизаторе R1 в 13:38:39 (около 13:40) и другие сообщения на маршрутизаторе R2 примерно в 9:45, он никак их не связывает и не понимает, что они свидетельствуют о той же проблеме.

Пример 23.10. Регистрационные сообщения от двух маршрутизаторов

```
*Oct 19 13:38:37.568: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on
Serial0/0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Oct 19 13:38:39.568: %LINK-5-CHANGED: Interface Serial0/0/0, changed
state to administratively down
*Oct 19 13:38:40.568: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/0, changed state to down

!=====================================================================

! Эти сообщения отправлены маршрутизатором R2
Oct 19 09:44:09.027: %LINK-3-UPDOWN: Interface Serial0/0/1, changed state
to down
Oct 19 09:44:09.027: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on
Serial0/0/1 from FULL to DOWN, Neighbor Down: Interface down or detached
Oct 19 09:44:10.027: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/1, changed state to down
```

На самом деле сообщения в обеих частях рис. 23.12 передаются в течение 0,5 секунды одно после другого, но часы этих двух маршрутизаторов не были синхронизированы. При синхронизации часов эти два маршрутизатора установили бы практически одинаковую временную метку для сообщений, существенно облегчив впоследствии чтение и сопоставление сообщений. Для проблем безопасности точные временные метки также важны, они позволяют сопоставлять регистрационные сообщения маршрутизатора и коммутатора с другими событиями, зарегистрированными другим программным обеспечением и оборудованием системы безопасности, что улучшает защиту против атак.

Чтобы настроить маршрутизатор или коммутатор на синхронизацию часов с существующим сервером NTP, достаточно одной команды конфигурации. Пример 23.11 демонстрирует команду `ntp server` на маршрутизаторе R1. Эта команда не заставит локальное устройство действовать как сервер NTP; она только укажет IP-адрес сервера NTP и заставит локальное устройство действовать как клиент NTP. В данном случае сервер имеет IP-адрес 172.16.2.2.

Пример 23.11. Настройка и проверка клиента NTP

```
R1# configure terminal
R1(config)# ntp server 172.16.2.2 version 4
R1(config)# ^Z
R1#

R1# show ntp status
Clock is synchronized, stratum 8, reference is 172.16.2.2
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is
2**21
ntp uptime is 4700 (1/100 of seconds), resolution is 4000
reference time is D42BD899.5FFCE014 (13:48:09.374 UTC Fri Oct 19 2012)
clock offset is -0.0033 msec, root delay is 1.28 msec
root dispersion is 3938.51 msec, peer dispersion is 187.59 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000
s/s
system poll interval is 64, last update was 42 sec ago.

R1# show ntp associations

address         ref clock       st  when  poll  reach  delay  offset  disp
*172.16.2.2     127.127.1.1     7   36    64    1      1.261  -0.001  7937.5
* sys.peer, # selected, + candidate, - outlyer, x falseticker, configured
```

Вторая часть примера демонстрирует две команды проверки NTP. В первой строке вывода команды `show ntp status` отображается состояние протокола NTP. В данном случае маршрутизатор R1 синхронизировал свое время с устройством 172.16.2.2. Эта команда выводит также текущее время и дату для часового пояса, заданного на маршрутизаторе. Команда `show ntp associations` выводит по одной строке для каждого устройства NTP, с которым связан маршрутизатор.

Существует и много других вариантов настройки NTP, кроме самой простой, представленной в примере 23.11. Например, устройства могут быть равноправными, влияя на время друг друга. Фактически маршрутизаторы и коммутаторы могут стать серверами NTP — достаточно одной команды (`ntp master`). Для настройки NTP также применима аутентификация, чтобы умышленный перевод часов маршрутизатора или коммутатора, а следовательно, и изменение временны*х* меток его сообщений не затруднили обнаружение атаки.

Обзор

Резюме

- Как и стандартные списки, расширенные списки управления доступом применяются для пакетов, либо входящих на интерфейс, либо исходящих из интерфейса. Система IOS проводит поиск в этом списке последовательно. Расширенные списки доступа также используют логику первого соответствия, поскольку маршрутизатор останавливает поиск по списку, как только обнаруживается первый соответствующий оператор, и предпринимает определенное в нем действие.
- Один оператор расширенного списка ACL может задать проверку нескольких элементов заголовка пакета, требуя точного соответствия всех параметров правилам данного оператора ACL. Такая мощная логика распознавания делает расширенные списки управления доступом и более полезными, и более сложными, чем стандартные списки ACL.
- Подобно стандартному нумерованному списку ACL IP, расширенный нумерованный список ACL IP также использует глобальную команду `access-list`. Синтаксис тот же, по крайней мере, в использовании ключевых слов `permit` и `deny`. Список параметров распознавания команд, конечно, отличается.
- "В частности, расширенная команда ACL `access-list` требует трех параметров соответствия: тип протокола IP, IP-адрес отправителя и IP-адрес получателя.
- Для типа протокола используется такое ключевое слово, как `tcp`, `udp` или `icmp`, для пакетов IP, у которых после заголовка IP есть заголовок TCP, UDP или ICMP соответственно. Либо можно использовать ключевое слово `ip`, означающее "все пакеты IP".
- В команде `access-list` расширенного списка ACL, чтобы пакет считался соответствующим команде, все параметры распознавания должны соответствовать полям пакета.
- Расширенные списки управления доступом позволяют также исследовать части заголовков TCP и UDP, в частности, поля номера порта получателя и отправителя. Номера портов идентифицируют приложение, которое посылает или получает данные.
- Когда команда расширенного списка ACL включает ключевое слово `tcp` или `udp`, она может (необязательно) проверить порт отправителя и/или получателя. Для этого синтаксис использует для номеров портов ключевые слова *равно* (`equal`), *не равно* (`not equal`), *меньше* (`less`), *больше* (`greater`) и *диапазон* (`range`). Кроме того, команда может использовать литеральные или десятичные номера портов либо более удобные ключевые слова для некоторых общеизвестных портов приложений.

- Настройка конфигурации списка управления доступом осуществляется с помощью команд `access-list`, а по завершении, чтобы задействовать список ACL, используется такая же команда `ip access-group`, применяемая для стандартных списков управления доступом. Все эти этапы отражают то, что происходит со стандартными списками управления доступом; однако при настройке помните о следующих различиях.
  - Располагайте расширенные списки управления доступом как можно ближе к отправителю пакетов, подлежащих фильтрации. Применение фильтрации ближе к источнику экономит полосу пропускания.
  - Помните, что пакет считается соответствующим оператору `access-list` только при полном совпадении всех параметров в одной из команд `access-list` с соответствующими полями пакета.
  - Для расширенной команды `access-list` могут использоваться номера 100–199 или 2000–2699, причем ни один номер не рассматривается как более предпочтительный.
- Именованные списки ACL IP имеют много сходств с нумерованными списками ACL IP. Они применяются для фильтрации пакетов, а также для многих других целей. Точно так же, подобно стандартным и расширенным нумерованным спискам ACL, которые отличаются возможностями распознавания пакетов, именованные списки ACL могут быть стандартными и расширенными.
  - Единственная действительно новая часть конфигурации именованных списков ACL — это глобальная команда конфигурации `ip access-list`, которая определяет, является ли список ACL стандартным или расширенным, а также задает имя.
  - Именованные списки ACL позволяют пользователю удалять и добавлять новые строки из режима конфигурации ACL.
- Средства редактирования ACL используют порядковые номера, добавляемые к каждому оператору `permit` или `deny` списка ACL, а также номера, представляющие последовательности операторов в списке ACL. Порядковые номера ACL предоставляют следующие средства и для нумерованных, и для именованных списков ACL.
  - **Новый стиль конфигурации для нумерованных списков**. Нумерованные списки ACL используют теперь стиль конфигурации, как у именованных, а также традиционный стиль; для расширенного редактирования списков ACL необходим новый стиль.
  - **Удаление отдельных строк**. С помощью команды `no` *`порядковый_номер`* можно удалять в списке управления доступом отдельные строки операторов `permit` или `deny`.
  - **Вставка новых строк**. Добавляемые команды `permit` и `deny` можно задавать в конфигурации с указанием порядкового номера, определяя местонахождение оператора в списке управления доступом.

- **Автоматическая нумерация**. Операционная система IOS сама добавляет командам порядковые номера при настройке, даже если они неизвестны инженеру.

- Когда внешний пользователь подключается к маршрутизатору или коммутатору по Telnet или SSH, операционная система IOS использует канал vty для предоставления ему пользовательского соединения. Применив списки ACL к этим входящим соединениям, операционная система IOS может фильтровать адреса хостов IPv4, от которых маршрутизаторы или коммутаторы могут принимать соединения telnet или SSH.

  В команде `access-class 3 in`, например, ключевое слово `in` указывает, что речь идет о входящих запросах на создание соединений Telnet и SSH с маршрутизатором, иными словами, о подключении к маршрутизатору с помощью этого протокола удаленного соединения.

- Ниже приведены основные правила списков ACL.
  - Размещайте расширенные списки управления доступом как можно ближе к отправителю пакета, чтобы сразу же отбрасывать определенные типы пакетов.
  - Размещайте стандартные списки управления доступом как можно ближе к получателю пакетов, поскольку они часто уничтожают пакеты, которые не должны быть уничтожены, если находятся ближе к отправителю.
  - Размещайте более специфичные (т.е. узкие) правила проверки ближе к началу списка управления доступом.
  - Прежде чем вносить изменения в список управления доступом, удалите список управления доступом на том интерфейсе, на котором он задан (с помощью команды `no ip access-group`).

- Протокол синхронизации сетевого времени (NTP) позволяет любому устройству, такому как маршрутизатор или коммутатор, синхронизировать свои часы. Если часы всех сетевых устройств синхронизированы, то сообщения могут быть отсортированы и просмотрены по дате и времени, что ускоряет и упрощает поиск причин произошедшего.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Для каких из следующих полей не может быть проведено сравнение на основе расширенного списка управления доступом IP? (Выберите два ответа.)

   А) Протокол.

   Б) IP-адрес отправителя.

   В) IP-адрес получателя.

   Г) Байт TOS.

   Д) URL.

   Е) Имя файла для передачи по протоколу FTP.

2. Какая из следующих команд `access-list` разрешает передачу пакетов от хоста 10.1.1.1 на все веб-серверы, IP-адреса которых начинаются с октетов 172.16.5? (Выберите два ответа.)

   А) `access-list 101 permit tcp host 10.1.1.1 172.16.5.0 0.0.0.255 eq www.`

   Б) `access-list 1951 permit ip host 10.1.1.1 172.16.5.0 0.0.0.255 eq www.`

   В) `access-list 2523 permit ip host 10.1.1.1 eq www 172.16.5.0 0.0.0.255.`

   Г) `access-list 2523 permit tcp host 10.1.1.1 eq www 172.16.5.0 0.0.0.255.`

   Д) `access-list 2523 permit tcp host 10.1.1.1 172.16.5.0 0.0.0.255 eq www.`

3. Какая из следующих команд `access-list` разрешает передачу пакетов любому веб-клиенту ото всех веб-серверов сети, IP-адреса которых начинаются с октетов 172.16.5?

   А) `access-list 101 permit tcp host 10.1.1.1 172.16.5.0 0.0.0.255 eq www.`

   Б) `access-list 1951 permit ip host 10.1.1.1 172.16.5.0 0.0.0.255 eq www.`

   В) `access-list 2523 permit tcp any eq www 172.16.5.0 0.0.0.255.`

   Г) `access-list 2523 permit tcp 172.16.5.0 0.0.0.255 eq www 172.16.5.0 0.0.0.255.`

   Д) `access-list 2523 permit tcp 172.16.5.0 0.0.0.255 eq www any.`

4. Для каких из следующих полей может быть проведено сравнение с использованием именованного расширенного списка управления доступом IP, но не нумерованного расширенного списка управления доступом IP?

   А) Протокол.

   Б) IP-адрес отправителя.

   В) IP-адрес получателя.

   Г) Байт TOS.

   Д) Все приведенные выше ответы не правильные.

5. В маршрутизаторе, работающем под управлением последней версии операционной системы IOS (15.0), инженер должен удалить вторую строку в списке управления доступом ACL 101, в конфигурацию которого в настоящее время входят четыре команды. Какие из следующих вариантов могут использоваться для этого? (Выберите два ответа.)

   А) Удаление всего списка управления доступом и повторный ввод в конфигурацию трех операторов ACL, которые должны остаться в списке управления доступом.

   Б) Удаление одной строки из списка управления доступом с помощью команды `no access-list...global`.

В) Удаление одной строки из списка управления доступом за счет перехода в режим настройки конфигурации списка применительно к данному списку управления доступом и последующего удаления только второй строки с указанием ее порядкового номера.

Г) Удаление последних трех строк из списка управления доступом в режиме глобальной конфигурации и добавление в дальнейшем двух последних операторов снова в список управления доступом.

6. Какими общими рекомендациями следует руководствоваться при использовании расширенных списков управления доступом IP?

А) Выполнять всю фильтрацию на выходе устройства, если это вообще возможно.

Б) Помещать более общие операторы ближе к началу списка управления доступом.

В) Фильтровать пакеты на устройстве, расположенном как можно ближе к устройству-отправителю.

Г) Упорядочивать команды ACL с учетом IP-адреса отправителя, от самых низких номеров к самым высоким, для повышения производительности.

7. Что из следующего истинно о функции клиента NTP на маршрутизаторе Cisco?

А) На основании сервера NTP клиент синхронизирует свой системный таймер.

Б) Он подсчитывает циклы процессора локального маршрутизатора, чтобы точнее учитывать время.

В) На основании сервера NTP клиент синхронизирует тактовую частоту своего последовательного канала.

Г) Клиент должен быть подключен к той же подсети, что и сервер NTP.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 23.6.

Таблица 23.6. Ключевые темы главы 23

Элемент	Описание	Страница
Рис. 23.3	Синтаксис расширенных списков ACL с необходимыми полями	667
Параграф	Важнейшая особенность логики расширенных списков ACL	668
Рис. 23.4	Поля номеров портов в заголовке TCP, сопровождающем заголовок IP	668
Рис. 23.5	Расширенный синтаксис команд ACL с номерами портов при использовании протокола TCP или UDP	669
Рис. 23.7	Фильтрация пакетов по данным о номере порта отправителя	670
Список	Советы и рекомендации по проверке портов TCP и UDP с использованием списков ACL	672
Список	Различия между именованными и нумерованными списками ACL	676
Список	Средства, предоставляемые операционной системой IOS 12.3 порядковыми номерами ACL	678
Список	Рекомендации по реализации списков ACL	686

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

Расширенный список управления доступом (extended access list), именованный список управления доступом (named access list), протокол синхронизации сетевого времени (Network Time Protocol — NTP)

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 23.7 приведен список команд конфигурации, а в табл. 23.9 приведены пользовательские команды главы. Фактически команды стоит запомнить, чтобы лучше понять материал главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 23.7. Команды конфигурации главы 23

Команда	Описание
`access-list` *номер_списка_управления_доступом* {`deny` | `permit`} *протокол отправитель шаблон_маски_отправителя получатель шаблон_маски_получателя* [`log`]	Глобальная команда для расширенных пронумерованных списков управления доступом. Используются номера 100–199 или 2000–2699 включительно
`access-list` *номер_списка_управления_доступом* {`deny` | `permit`} `tcp` *отправитель шаблон_маски_отправителя* [*оператор* [*порт*]] *получатель шаблон_маски_получателя* [*оператор* [*порт*]] [`log`]	Версия команды `access-list` с параметрами, специфичными для протокола TCP
`access-list` *номер_списка_управления_доступом* `remark` *текст*	Задает комментарий, помогающий запомнить, для чего предназначен список управления доступом
`ip access-group` {*номер* | *имя* [`in` | `out`]}	Подкоманда интерфейса, применяющая списки управления доступом
`access-class` *номер* | *имя* [`in` | `out`]	Команда режима конфигурирования линий, позволяющая включить стандартные или расширенные списки управления доступом для них
`ip access-list` {`standard` | `extended`} *имя*	Глобальная команда настройки именованного стандартного или расширенного списка ACL с переходом в режим настройки списка ACL
{`deny` | `permit`} *отправитель* [*шаблон_маски_отправителя*] [`log`]	Подкоманда режима ACL, предназначенная для ввода в конфигурацию сведений о правилах проверки и действиях, относящихся к стандартному именованному списку ACL

Окончание табл. 23.7

Команда	Описание
{deny | permit} *протокол отправитель шаблон_маски_отправителя получатель шаблон_маски_получателя* [log]	Подкоманда режима ACL, предназначенная для ввода в конфигурацию сведений о правилах проверки и действиях, относящихся к расширенному именованному списку ACL
{deny | permit} tcp *отправитель шаблон_маски_отправителя* [*оператор* [*порт*]] *получатель шаблон_маски_получателя* [*оператор* [*порт*]] [log]	Подкоманда режима ACL, предназначенная для ввода в конфигурацию сведений о критериях сопоставления и действиях, относящихся к именованному списку управления доступом, который сопоставляется с сегментами TCP
remark *текст*	Подкоманда режима ACL, предназначенная для ввода в конфигурацию описания именованного списка ACL

Таблица 23.8. Команды настройки защиты устройства главы 23

Команда	Описание
enable secret *пароль*	Глобальная команда, задающая пароль коммутатора, необходимый любому пользователю для доступа к привилегированному режиму
enable password *пароль*	Глобальная команда, задающая пароль коммутатора, необходимый любому пользователю для доступа к привилегированному режиму
login local	Режим конфигурации консоли и vty. Указывает IOS запрашивать имя пользователя и пароль перед вводом локальных и глобальных команд конфигурации username на данном коммутаторе или маршрутизаторе
username *имя* secret *пароль*	Глобальная команда, определяющая одно из нескольких возможных имен пользователя и связанных с ним паролей, используемых для аутентификации. Используется при применении команды конфигурации login local
crypto key generate rsa	Глобальная команда. Создает и сохраняет (в скрытой области флеш-памяти) ключи, необходимые для SSH
transport input {telnet | ssh | all | none}	Режим конфигурации линии vty. Определяет, разрешен ли доступ по протоколу Telnet и/или SSH к этому коммутатору. Оба значения могут быть заданы в одной команде, чтобы разрешить доступ и Telnet, и SSH (стандартное значение)
service password-encryption	Глобальная команда, шифрующая (слабо) открытые пароли, возможно, определенные по имени пользователя, пароли привилегированного режима и пароли команд линии
[no] ip http server	Глобальная команда, включающая и отключающая (с параметром no) службы HTTP, поддерживающей интерфейс GUI маршрутизатора или коммутатора

Окончание табл. 23.8

Команда	Описание
`[no] service tcp-small-servers`	Глобальная команда, включающая и отключающая (с параметром no) группу малых служб TCP (echo, discard, daytime и charge)
`[no] service udp-small-servers`	Глобальная команда, включающая и отключающая (с параметром no) группу малых служб UDP (echo, discard, daytime и charge)
`[no] cdp run`	Глобальная команда, включающая и отключающая (с параметром no) протокол CDP для всего маршрутизатора или коммутатора
`[no] cdp enable`	Глобальная команда, включающая и отключающая (с параметром no) протокол CDP для данного интерфейса
`ntp server` *IP-адресс_сервера* `[version 1..4]`	Настраивает маршрутизатор или коммутатор так, чтобы он действовал как клиент NTP и использовал указанный сервер NTP

Таблица 23.9. Пользовательские команды главы 23

Команда	Описание
`show ip` *интерфейс* `[`*тип номер*`]`	Выводит информацию о списках управления доступом, примененных на интерфейсе
`show access-lists [`*номер_списка_управления_доступом	имя_списка*`]`	Выводит сведения о введенных в конфигурацию списках управления доступом для всех протоколов
`show ip access-lists [`*номер_списка_управления_доступом	имя_списка*`]`	Выводит сведения о списках управления доступом IP
`show ntp status`	Выводит несколько строк информации о состоянии протокола NTP, включая IP-адреса всех равноправных узлов NTP
`show ntp associations`	Выводит одну строку, идентифицирующую равный маршрутизатор, связанный с данным по протоколу NTP

Практика

Ответы на практические задания главы

В табл. 8.10 содержатся ответы на практические задания, приведенные в табл. 8.6. Обратите внимание на то, что для любого вопроса, упоминающего клиент, можно выбирать соответствие порту, номер которого больше 1023. Ответы в данной таблице игнорируют эту возможность, но только для демонстрации одного из примеров, в ответе на первую задачу приведен вариант со ссылкой на порт клиента с номером больше 1023 и вариант без нее. Остальные ответы просто опускают эту часть логики.

Таблица 23.7. Создание однострочных расширенных списков ACL. Ответы

	Критерий
1	`access-list 101 permit tcp host 10.1.1.1 10.1.2.0 0.0.0.255 eq www` `access-list 101 permit tcp host 10.1.1.1 gt 1023 10.1.2.0 0.0.0.255 eq www`
2	`access-list 102 permit tcp 172.16.4.0 0.0.0.127 172.16.3.0 0.0.0.127 eq telnet`
3	`access-list 103 permit icmp 192.168.7.192 0.0.0.63 192.168.7.8 0.0.0.7`
4	`access-list 104 permit tcp 10.2.2.0 0.0.1.255 eq www 10.4.4.0 0.0.3.255`
5	`access-list 105 permit tcp 172.20.1.0 0.0.0.255 eq 23 172.20.44.0 0.0.1.255`
6	`access-list 106 permit tcp 192.168.99.96 0.0.0.15 192.168.176.0 0.0.0.15 eq www`
7	`access-list 107 permit icmp 10.55.66.0 0.0.0.127 10.66.55.0 0.0.0.63`
8	`access-list 108 permit ip any any`

Ответы на контрольные вопросы:

1 Д и Е. **2** А и Д. **3** Д. **4** Д. **5** А и В. **6** В. **7** А.

Глава 24

Трансляция сетевых адресов

В этой последней главе о протоколе IPv4 рассматривается весьма популярная и очень важная часть как корпоративной сети, так и сети малого или домашнего офиса (SOHO): трансляция сетевых адресов, или NAT. Технология NAT позволила решить наибольшую проблему протокола IPv4: к середине 1990-х годов пространство IPv4-адресов могло быть полностью исчерпано. Если бы это произошло, рост Интернета оказался бы невозможен, а его разработка значительно замедлилась бы.

Фактически в этой главе обсуждаются два краткосрочных решения проблемы исчерпания IPv4-адресов, что послужит также хорошим предисловием к обсуждению протокола IP версии 6 (IPv6). Технология NAT, наряду с бесклассовой междоменной маршрутизацией (CIDR), смогла продлить жизнь протокола IPv4 как протокола сетевого уровня Интернета с 1990-х до 2010-х годов. Долгосрочное решение, которое станет стандартом для Интернета (IPv6), обсуждается в части VII этой книги.

Эта глава разделена по темам на три главных раздела. В первом разделе обсуждается проблема исчерпания пространства IPv4-адреса, вызванная революцией Интернета в 1990-х годах. Во втором разделе рассматриваются фундаментальная концепция NAT, несколько разновидностей применения NAT, а также такое средство, как *трансляция адресов с использованием портов* (Port Address Translation — PAT), позволяющее экономить IPv4-адреса. В заключительном разделе демонстрируется настройка NAT в интерфейсе командной строки Cisco (CLI) операционной системы IOS, а также поиск и устранение неисправностей NAT.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Передача данных между двумя хостами по сети.

Службы IP

Базовые операции NAT.

Цель.

Пул.

Статический.

1 к 1.

Перегрузка.

Исходная адресация.

Односторонний NAT.

Настройка и проверка NAT для заданных требований сети.

Основные темы

Перспективы масштабируемости адресов протокола IPv4

При первоначальном проектировании Интернета предполагалось, что каждая организация запрашивает и получает один или несколько зарегистрированных классовых номеров сети (IP-адресов). Администраторы программы следили за тем, чтобы ни один из адресов сетей IP не повторялся. До тех пор, пока каждая организация использовала IP-адреса только из зарегистрированного для нее диапазона, они не повторялись и маршрутизация IP работала без проблем.

В течение определенного периода подключение к Интернету только через один или несколько зарегистрированных сетевых адресов функционировало благополучно. Однако уже в начале 1990-х годов стало очевидно, что Интернет растет столь быстро, что уже к середине 1990-х годов все адреса сетей IP будут исчерпаны (присвоены). Возникли опасения, что доступные адреса будут полностью исчерпаны и некоторые организации уже не смогут подключиться к Интернету.

Главным долгосрочным решением проблемы масштабируемости IP-адресов могло бы стать только увеличение размера IP-адреса. Один этот факт был наиболее существенной предпосылкой появления версии 6 протокола IP (IPv6) (версия 5 появилась значительно раньше, однако так и не была применена, поэтому следующая версия поучила номер 6). В протоколе IPv6 используется 128-битовый адрес вместо 32-битового в протоколе IPv4. Используя прежний или улучшенный процесс назначения уникальных диапазонов адресов каждой организации, подключенной к Интернету, протокол IPv6 может без проблем обеспечивать доступ к Интернету всех организаций и отдельных пользователей планеты, поскольку количество возможных адресов протокола IPv6 теоретически достигает 10^{38}.

Были также предложены некоторые краткосрочные решения данной проблемы, однако для практических целей совместно использовались три стандарта. Два из них тесно связаны между собой — *трансляция сетевых адресов* (Network Address Translation — NAT) и частная адресация. Совместно эти средства позволяют использовать незарегистрированные сетевые адреса протокола IP и вместе с тем оставляют возможность выхода в Интернет. Третий стандарт, *бесклассовая междоменная маршрутизация* (Classless Interdomain Routing — CIDR), позволяет провайдерам ISP уменьшить количество нерационально расходуемых IP-адресов за счет назначения компании подмножества сетевых адресов вместо целой сети. Маршрутизация CIDR также позволяет провайдерам суммировать (обобщать) маршруты, соответствующие нескольким сетям классов A, B и C, в отдельные маршруты, сокращая размер таблиц маршрутизации Интернета.

ВНИМАНИЕ!

Эти средства работали хорошо, однако в начале 1990-х предполагалось, что мир исчерпает IPv4-адреса к середине 1990-х, но на февраль 2011 года организация IANA еще не исчерпала пространство IPv4-адресов. См. страницу на веб-сайте автора, посвященную этой главе и содержащую некоторые ссылки на соответствующие статьи.

Маршрутизация CIDR

Маршрутизация CIDR представляет собой глобальное соглашение о назначении адресов, которое определяет, каким образом Агентство по назначению адресов Интернета (Internet Assigned Numbers Authority — IANA), его филиалы (member agencies) и провайдеры ISP назначают глобально уникальные адресные пространства протокола IPv4 отдельным организациям.

У бесклассовой междоменной маршрутизации, определенной в документе RFC 4632, есть две основные задачи. Во-первых, бесклассовая междоменная маршрутизация определяет способ присвоения открытых IP-адресов во всем мире, чтобы обеспечить объединение или суммирование маршрутов. Суммирование маршрутов существенно сокращают размер таблиц маршрутизации. Во-вторых, бесклассовая междоменная маршрутизация определяет правила, позволяющие провайдерам ISP присваивать открытые IP-адреса не только блоками размером во всю сеть класса A, B или C. Бесклассовая междоменная маршрутизация позволяет провайдерам услуг Интернета присвоить блок открытых IPv4-адресов такого размера, который лучше удовлетворяет потребности конкретного клиента.

Агрегирование маршрутов для уменьшения размера таблиц маршрутизации

Представим себе маршрутизатор Интернета, который подключен ко всем имеющимся на планете сетям классов A, B и C! Ведь существует более двух миллионов сетей класса C! Если бы маршрутизаторам Интернета понадобилось хранить в своих таблицах все возможные классовые сети, то им потребовался бы огромный объем памяти, а поиск в этих таблицах потребовал бы большой вычислительной мощности.

Бесклассовая междоменная маршрутизация определяет стратегию объединения или суммирования маршрутов к открытым IPv4-адресам в Интернете. Эта стратегия полагается на международную стратегию присвоения IPv4-адресов и несложный математический механизм, позволяя заменить множество маршрутов для меньших диапазонов адресов одним маршрутом для большего диапазона адресов.

На рис. 24.1 показан типичный случай того, как бесклассовая междоменная маршрутизация могла бы использоваться для замены более чем 65 000 маршрутов одним. Предположим, например, что провайдер ISP 1 владеет сетями класса C от 198.0.0.0 до 198.255.255.0 (как ни странно, но это вполне допустимые номера сетей класса C). Другими словами, агентство IANA присваивало все адреса, начинающиеся на 198, одному из пяти региональных представительств, а оно присвоило весь этот диапазон одному крупному провайдеру ISP. Такая концепция присвоения является частью бесклассовой междоменной маршрутизации, поскольку на ней основано суммирование маршрутов.

Бесклассовая междоменная маршрутизация определяет, как создать один маршрут для всех 2^{16} сетей класса C, начинающихся на 198. На рис. 24.1, *слева*, приведены провайдеры услуг Интернета с одним маршрутом к сети 198.0.0.0/8 каждый, другими словами, маршрутом ко всем хостам, IP-адрес которых начинается на 198. На 198 начинается 65 536 сетей IP класса C, и один суммарный маршрут представляет все эти сети IP.

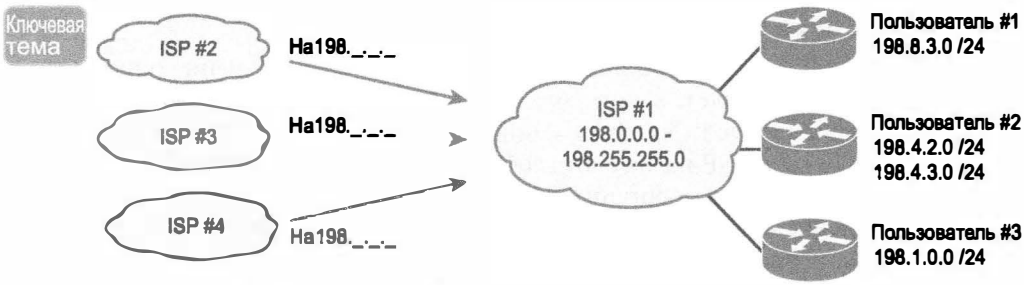


Рис. 24.1. Типичный случай использования маршрутизации CIDR

Бесклассовая междоменная маршрутизация требует использования бесклассового протокола маршрутизации, который, по определению, посылает наряду с каждым маршрутом маску. Бесклассовые протоколы маршрутизации рассматривают каждый маршрут скорее как математическую задачу, игнорируя правила класса A, B и C. Например, сеть 198.0.0.0/8 (198.0.0.0, маска 255.0.0.0) определяет набор адресов, первые 8 битов которых равны десятичному 198. Провайдер ISP 1 анонсирует этот маршрут другим провайдерам услуг Интернета, которые нуждаются только в маршруте к сети 198.0.0.0/8. На своих маршрутизаторах провайдер ISP 1 знает, какие сети класса C каким клиентским площадкам принадлежат. Именно так бесклассовая междоменная маршрутизация обеспечивает маршрутизаторам Интернета большую масштабируемость таблиц маршрутизации за счет сокращения количества записей.

Сохранение адресов протокола IPv4

Маршрутизация CIDR позволяет также уменьшить вероятность исчерпания IPv4-адресов и позволить провайдеру ISP выделить подмножество адресов сетей классов A, B и C отдельному пользователю. Предположим, например, что пользователю 1 провайдера ISP 1 требуется только десять IP-адресов, а пользователю 3 — 25 IP-адресов. Провайдер ISP 1 выполняет такие действия: назначает пользователю 1 подсеть IP 198.8.3.16/28 с возможными адресами от 198.8.3.17 до 198.8.3.30. Для пользователя 3 провайдер ISP 1 предлагает подсеть 198.8.3.32/27 с 30 возможными адресами (198.8.3.33–198.8.3.62). Провайдер ISP удовлетворил потребности пользователей и при этом использовал не все адреса сети 198.8.3.0 класса C.

Маршрутизация CIDR предотвращает нерациональную трату IP-адресов, уменьшая потребность в зарегистрированных IP-адресах. Вместо того чтобы два пользователя полностью потребляли два диапазона сетевых адресов, каждый из них использует лишь небольшую часть одной сети класса C. В то же время маршрутизация CIDR наряду с разумным администрированием последовательных сетевых адресов каждому провайдеру ISP позволяет поддерживать на маршрутизаторах Интернета таблицы маршрутизации значительно меньшего размера, чем потребовалось бы в ином случае.

Частные адреса

Некоторые компьютеры, вероятно, никогда не будут подключаться к Интернету непосредственно. IP-адреса этих компьютеров могут быть дубликатами зарегистрированных IP-адресов в Интернете. При проектировании IP-адресации для такой се-

ти организация может выбрать и использовать произвольные сетевые адреса, и все будет прекрасно работать. Например, можно приобрести несколько маршрутизаторов, соединить их в своем офисе, задать IP-адреса из сети 1.0.0.0, и все будет работать. Используемые IP-адреса могут быть дубликатами реальных IP-адресов в Интернете, но если они используются только для внутренних целей (в лаборатории, офисе), то никаких проблем не возникнет.

При создании частной сети, не имеющей соединения с Интернетом, можно использовать сетевые IP-адреса, называемые *частными интерсетями* (private internet), как это определено в документе RFC 1918 "Выделение адресов для частных интерсетей" (Address Allocation for Private Internets). В этом документе RFC определено множество сетей, которые никогда не будут назначены какой-либо организации в качестве зарегистрированного сетевого номера. Вместо чьих-либо зарегистрированных сетевых адресов можно использовать номера из диапазона, которые никем не используются в открытой сети Интернет. В табл. 24.1 приведено пространство частных адресов, определенное в документе RFC 1918.

Таблица 24.1. Пространство частных адресов согласно документу RFC 1918

Ключевая тема

Диапазон IP-адресов	Класс сети	Количество сетей
10.0.0.0–10.255.255.255	A	1
172.16.0.0–172.31.255.255	B	16
192.168.0.0–192.168.255.255	C	256

Иными словами, эти адреса сетей могут использоваться любой организацией. Однако ни одна организация не имеет права анонсировать эти сети, используя протокол маршрутизации в Интернете.

Может возникнуть вопрос: "Зачем беспокоиться о резервировании специальных частных сетевых адресов, если не имеет значения возможность их повторения?" Оказывается, что в своей внутренней сети можно применять частную адресацию и в то же время подключаться к Интернету, если используется *трансляция сетевых адресов* (Network Address Translation — NAT), которая рассматривается в этой главе.

Принципы трансляции сетевых адресов

Трансляция NAT, определенная в документе RFC 3022, позволяет хосту, не имеющему действительного, зарегистрированного глобально уникального IP-адреса, осуществлять связь с другими хостами через Интернет. Эти хосты могут использовать частные адреса или адреса, назначенные другим организациям. В любом из этих случаев трансляция NAT позволяет продолжать использование этих адресов, не пригодных для Интернета, и в то же время осуществлять связь с хостами в Интернете.

Трансляция NAT обеспечивает это за счет использования действительных зарегистрированных IP-адресов для представления данного частного адреса всем остальным хостам Интернета. Трансляция NAT заменяет частные IP-адреса открытыми зарегистрированными IP-адресами в каждом пакете протокола IP, как показано на рис. 24.2.

Ключевая тема

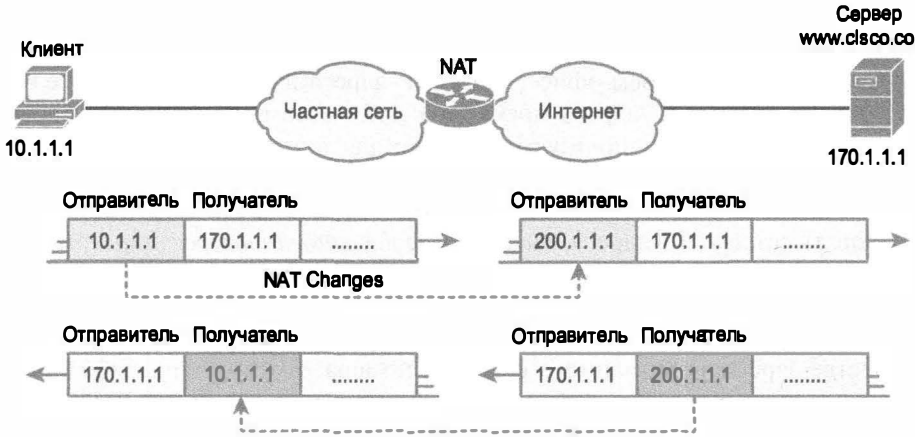


Рис. 24.2. Замена IP-адресов при использовании трансляции NAT: частная адресация

Обратите внимание на то, что, осуществляя трансляцию NAT, маршрутизатор изменяет IP-адрес отправителя в тот момент, когда пакет покидает организацию. Маршрутизатор, поддерживающий NAT, также изменяет адрес получателя каждого пакета, который возвращается назад в частную сеть (на рис. 24.2 это зарегистрированная сеть 200.1.1.0). Функция NAT, настроенная на маршрутизаторе с надписью "NAT", выполняет трансляцию адресов. Программное обеспечение Cisco IOS поддерживает несколько разновидностей трансляции NAT. Далее в настоящей главе описываются принципы, лежащие в основе этих разновидностей трансляции. В следующих разделах описывается настройка, связанная с каждой опциональной возможностью.

Статическая трансляция NAT

Статическая трансляция NAT работает точно так же, как было показано в примере на рис. 24.2, однако IP-адреса преобразуются друг в друга статически. Для того чтобы читателю стали понятными результаты статической трансляции NAT и для объяснения некоторых ключевых терминов, на рис. 24.3 приведен аналогичный пример с более подробным описанием.

Рис. 24.3. Трансляция NAT с внутренними локальными и глобальными адресами

Сначала рассмотрим основные принципы. Провайдер ISP компании назначает ей зарегистрированный номер сети 200.1.1.0. Соответственно маршрутизатор NAT должен сделать так, чтобы этот частный адрес выглядел таким образом, как если бы он находился в сети 200.1.1.0. Для этого маршрутизатор NAT изменяет IP-адрес отправителя в пакетах, которые на рисунке пересылаются слева направо.

В данном примере маршрутизатор NAT изменяет адрес отправителя 10.1.1.1 (на рисунке обозначен как "SA") на адрес 200.1.1.1. При использовании статической трансляции маршрутизатор NAT просто устанавливает взаимно однозначное соответствие между частным и зарегистрированным адресом, от имени которого он выступает. На маршрутизаторе NAT настроено статическое соответствие частного адреса 10.1.1.1 с открытым зарегистрированным адресом 200.1.1.1.

Поддержка двух хостов IP в частной сети требует второго взаимно однозначного сопоставления со вторым IP-адресом в диапазоне открытых адресов. Например, для поддержки адреса 10.1.1.2 маршрутизатор сопоставляет адрес 10.1.1.2 с адресом 200.1.1.2. Поскольку предприятие имеет одну зарегистрированную сеть класса C, используя трансляцию NAT, оно может поддерживать до 254 частных IP-адресов (при этом зарезервированы два обычных адреса — номер сети и ее широковещательный адрес).

Терминология, используемая для технологии NAT, особенно относящаяся к ее настройке, может показаться несколько непривычной. Обратите внимание на то, что в таблице NAT на рис. 24.3 частные IP-адреса приводятся как "частные", а открытые зарегистрированные адреса сети 200.1.1.0 — как "открытые". Компания Cisco использует термин *внутренний локальный адрес* (inside local) для частных IP-адресов (как это сделано в данном примере) и термин *внутренний глобальный адрес* (inside global) для открытых IP-адресов.

В терминологии Cisco корпоративная сеть, использующая частные адреса, а следовательно, требующая использования NAT, является внутренней частью сети. Подключенный к Интернету интерфейс трансляции NAT является внешней частью сети. Хост, которому необходима трансляция NAT (в данном примере 10.1.1.1), имеет IP-адрес, который он использует внутри сети, и ему требуется IP-адрес, который будет представлять его вне этой сети. Поскольку хосту фактически нужны два разных адреса для его представления, требуются два термина. В документации Cisco частные IP-адреса, используемые во внутренней сети, называются *внутренними локальными адресами* (inside local), а адреса, используемые для представления хоста в Интернете, — *внутренними глобальными адресами* (inside global). На рис. 24.4 приведен тот же пример с терминологией.

В большинстве типичных конфигураций NAT изменяется только IP-адрес внутренних хостов. Соответственно, в текущей таблице NAT, показанной на рис. 24.4, приведены внутренний локальный и соответствующий внутренний глобальный зарегистрированный адреса. Однако внешний IP-адрес хоста также может быть изменен с помощью NAT. Когда это происходит, термины "внешний локальный" и "внешний глобальный" означают IP-адреса, используемые для представления этого хоста во внутренней и внешней сети соответственно. В табл. 24.2 обобщены используемые термины и их определения.

Ключевая тема

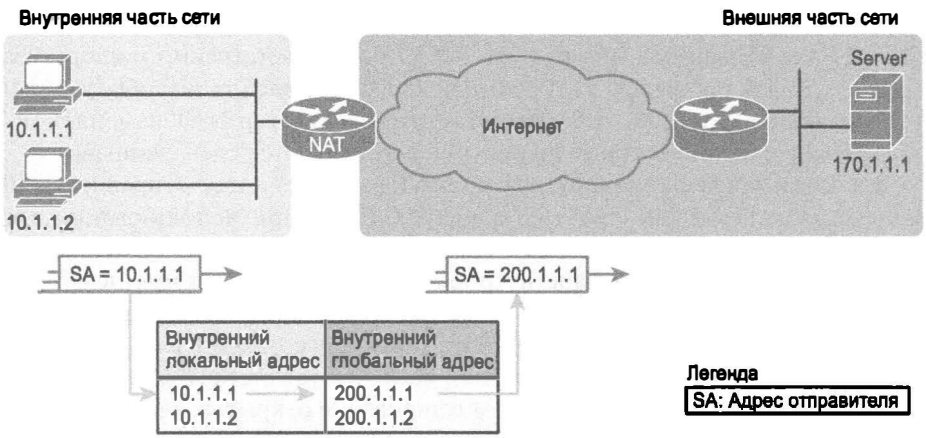


Рис. 24.4. Диаграмма типичной сети NAT и ключевые термины трансляции NAT

Ключевая тема

Таблица 24.2. Термины трансляции NAT и их определения

Термин	Определение
Внутренний локальный адрес (Inside local)	При типичном проектировании NAT термин "внутренний" относится к адресу, используемому для хоста на предприятии. Внутренним локальным называется действующий IP-адрес, назначенный хосту в частной сети предприятия. Более наглядным термином мог бы быть "внутренний частный", поскольку зачастую (но не всегда) внутренние адреса являются также частными
Внутренний глобальный адрес (Inside global)	При типичном проектировании NAT термин "внутренний" относится к адресу, используемому для хоста на предприятии. Трансляция NAT использует внутренний глобальный адрес для представления внутреннего хоста, когда пакет пересылается через внешнюю сеть, обычно через Интернет. Маршрутизатор NAT изменяет IP-адрес отправителя в пакете, посылаемом внутренним хостом, с внутреннего локального адреса на внутренний глобальный адрес, в то время, когда пакет пересылается из внутренней сети во внешнюю. Более наглядным мог бы быть термин "внутренний открытый (общедоступный)", поскольку при использовании на предприятии адресов RFC 1918 внутренний глобальный представляет внутренний хост с открытым IP-адресом, который может быть использован для маршрутизации в открытой сети Интернет
Внешний глобальный адрес (Outside global)	При типичном проектировании NAT термин "внешний" относится к адресу, используемому для хоста вне предприятия, иными словами — в Интернете. Внешний глобальный адрес представляет собой реальный IP-адрес, назначенный хосту, который находится в сети, обычно в Интернете. Более содержательным (точным) термином мог бы быть "внешний открытый", поскольку внешний глобальный адрес представляет внешний хост с открытым IP-адресом, который может использоваться для маршрутизации в открытой сети Интернет

Окончание табл. 24.2

Термин	Определение
Внешний локальный адрес (Outside local)	Технология NAT может транслировать внешние IP-адреса, т.е. IP-адреса, представляющие хост вне сети предприятия, хотя эта опциональная возможность не очень популярна. Когда маршрутизатор NAT пересылает пакет из внутренней сети во внешнюю, используя NAT для изменения внешнего адреса, IP-адрес, представляющий внешний хост в качестве IP-адреса получателя в заголовке пакета, называется внешним локальным IP-адресом. Более содержательным мог бы быть термин "внешний частный", поскольку, используя адреса RFC 1918 на предприятии, внешний локальный адрес представляет внешний хост с частным IP-адресом из RFC 1918

Динамическая трансляция NAT

В сравнении со статической динамическая трансляция NAT имеет как сходства, так и отличия. Как и в случае использования статической трансляции NAT, маршрутизатор NAT устанавливает взаимно однозначное соответствие между внутренним локальным и внутренним глобальным адресами и изменяет IP-адреса в пакетах, когда они входят во внутреннюю сеть и выходят из нее. Однако преобразование внутренних локальных адресов во внутренние глобальные адреса происходит динамически.

Динамическая трансляция NAT создает пул возможных внутренних глобальных адресов и определяет критерий соответствия для определения того, какие внутренние глобальные IP-адреса должны транслироваться с помощью NAT. Например, на рис. 24.5 был установлен пул из пяти глобальных IP-адресов в диапазоне 200.1.1.1–200.1.1.5. Трансляция NAT также настроена для преобразования всех внутренних локальных адресов, которые начинаются с октетов 10.1.1.

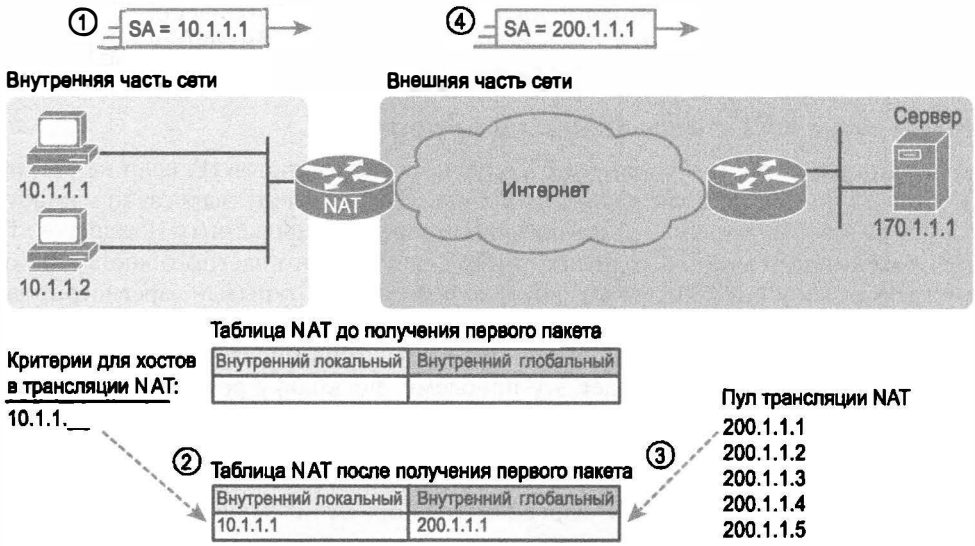


Рис. 24.5. Динамическая трансляция NAT

Номера 1, 2, 3 и 4 на рисунке относятся к такой последовательности событий.

1. Хост 10.1.1.1 посылает свой первый пакет на сервер, расположенный по адресу 170.1.1.1.
2. Когда данный пакет поступает на маршрутизатор NAT, он применяет логику проверки соответствия и решает, следует ли применить к этому пакету трансляцию NAT. Поскольку логика маршрутизатора настроена на соответствие IP-адресам отправителя, которые начинаются с 10.1.1, маршрутизатор добавляет запись в свою таблицу NAT для адреса 10.1.1.1 в качестве внутреннего локального адреса.
3. Маршрутизатору NAT требуется выделить IP-адрес из пула действительных внутренних глобальных адресов. Он выбирает первый доступный адрес (в данном случае 200.1.1.1) и добавляет его в таблицу NAT для завершения записи.
4. Маршрутизатор NAT транслирует IP-адрес отправителя и пересылает пакет. Динамическая запись остается в таблице до тех пор, пока продолжается передача данных. Можно задать значение тайм-аута, который определяет, как долго должен ожидать маршрутизатор, пока не транслируются пакеты с таким адресом, перед удалением этой динамической записи. Можно также вручную удалить из таблицы динамические ссылки с помощью команды `clear ip nat translation *`.

Трансляция NAT может быть настроена с большим количеством IP-адресов в списке внутренних локальных адресов, чем в пуле внутренних глобальных адресов. Маршрутизатор выделяет адреса из пула до тех пор, пока все они не будут выделены. Если поступает новый пакет от еще одного внутреннего хоста и ему требуется запись NAT, а все находящиеся в пуле IP-адреса уже используются, то маршрутизатор просто отбрасывает данный пакет. Пользователь должен повторять попытку до тех пор, пока не истечет время тайм-аута записи NAT; в этот момент функция NAT работает для следующего хоста, который отправляет пакет. По существу, размер внутреннего глобального пула адресов должен соответствовать максимальному количеству конкурирующих хостов, которым требуется одновременный доступ к Интернету (кроме случая использования трансляции PAT, который описан в следующем разделе).

Перезагрузка NAT с использованием портов

В некоторых сетях требуется, чтобы большинство их хостов IP, если не все, имели доступ к Интернету. Если такая сеть использует частные IP-адреса, то маршрутизатору NAT необходим очень большой набор зарегистрированных IP-адресов. При использовании статической трансляции NAT для каждого частного хоста IP, которому требуется доступ к Интернету, необходимо иметь открытый зарегистрированный IP-адрес, что полностью подрывает намерение уменьшить количество нужных организации открытых адресов протокола IPv4. Динамическая трансляция NAT в определенной степени смягчает эту проблему, поскольку редко требуется одновременный выход в Интернет всем хостам объединенной сети.

Однако если большей части хостов сети необходим доступ к Интернету в течение всего обычного рабочего дня, то трансляции NAT по-прежнему нужно большое количество зарегистрированных IP-адресов, что вновь не позволяет уменьшить расход адресов протокола IPv4.

Функция перезагрузки NAT, называемая также *трансляцией адресов портов* (Port Address Translation — PAT), решает эту проблему. Перезагрузка позволяет трансляции NAT выполнить масштабирование для поддержки многих клиентов с использованием всего лишь нескольких открытых IP-адресов.

Ключом к пониманию принципа является способ использования хостами портов TCP и UDP. Чтобы увидеть почему, рассмотрим сначала концепцию трех отдельных соединений TCP с веб-сервером, от трех разных хостов, как показано на рис. 24.6.

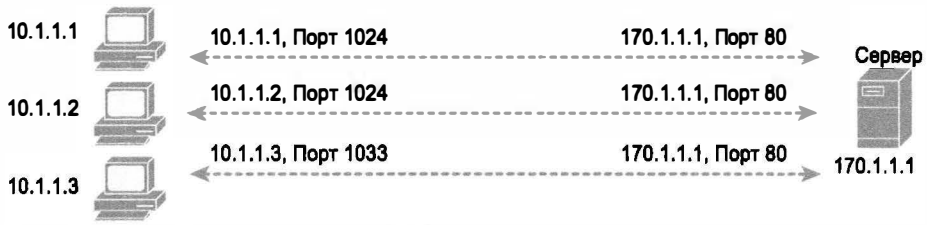


Рис. 24.6. Три соединения протокола TCP: от трех различных хостов и от одного хоста

Теперь сравните эти три соединения TCP (см. рис. 24.6) с тремя подобными соединениями TCP, но уже от одного клиента, как показано на рис. 24.7. Сервер понимает различие, поскольку он видит IP-адреса и номера портов TCP, используемые клиентами на обоих рисунках. Однако фактически его не заботит, исходят ли соединения TCP от разных хостов или от того же хоста; сервер только получает и посылает данные по каждому соединению.

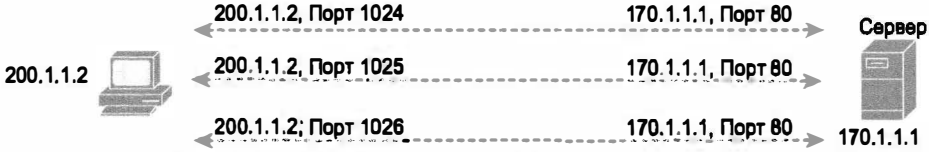


Рис. 24.7. Три соединения протокола TCP от одного компьютера

Трансляция NAT использует в своих интересах тот факт, что с точки зрения транспортного уровня сервер не заботит, имеет ли он по одному соединению с каждым из трех отдельных хостов или три соединения с одним хостом. Перегрузка NAT (PAT) преобразует не только адрес, но и при необходимости номер порта, делая многие потоки TCP или UDP от разных хостов похожими на то же количество потоков от одного хоста (рис. 24.8).

При создании динамического сопоставления PAT выбирает не только внутренний глобальный IP-адрес, но и уникальный номер порта, который будет использоваться с этим адресом. Маршрутизатор поддерживает запись в таблице NAT для каждой уникальной комбинации внутреннего локального адреса и порта; при этом поддерживается трансляция во внутренний глобальный адрес и уникальный номер порта, связанный с внутренним глобальным адресом. И поскольку поле номера порта состоит из 16 бит, перезагрузка NAT позволяет использовать более 65 тысяч номеров портов, что позволяет ей выполнять масштабирование без необходимости иметь много зарегистрированных IP-адресов (во многих случаях требуется лишь один внутренний глобальный IP-адрес).

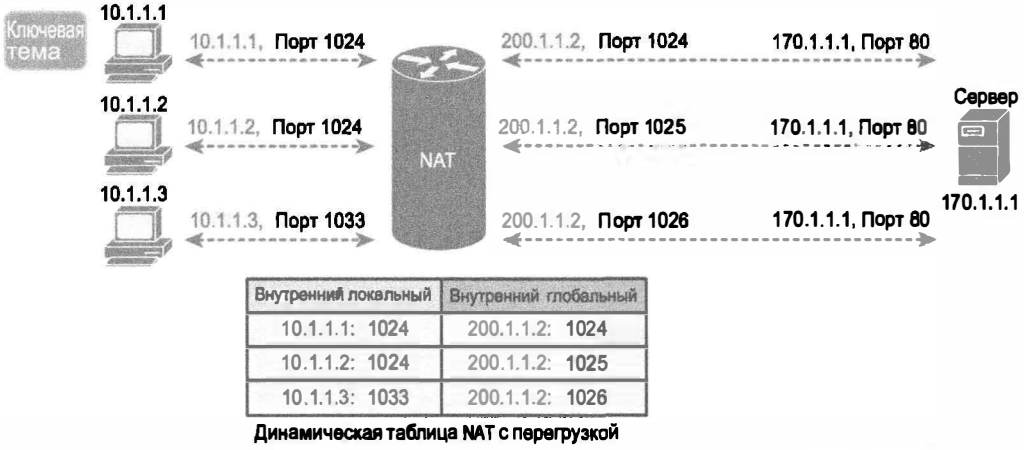


Рис. 24.8. Принципы сохранения адресов при перезагрузке NAT (PAT)

Из трех типов трансляции NAT, рассмотренных в данной главе, PAT, безусловно, является наиболее популярным. Как статическое, так и динамическое преобразование NAT требует взаимно однозначного сопоставления внутреннего локального и внутреннего глобального адресов. Трансляция PAT значительно уменьшает количество необходимых зарегистрированных IP-адресов по сравнению с двумя другими альтернативами NAT.

Перегрузка NAT (PAT) на маршрутизаторах потребительского класса

В следующем разделе этой главы будут описаны настройка и проверка перегрузки NAT (или PAT) на маршрутизаторе Cisco корпоративного класса. На маршрутизаторах Cisco потребительского класса многие средства включены изначально, в том числе PAT. Стратегия маршрутизаторов потребительского класса позволяет покупателю установить маршрутизатор физически, подключить кабели и использовать маршрутизатор без настройки. В данном разделе будет показано, как маршрутизаторы потребительского класса позволяют включить PAT при использовании первоначальных параметров.

Как упоминалось в главе 3, продаваемые в магазинах устройства под названием "маршрутизатор" фактически являются многофункциональными: это и маршрутизатор, и коммутатор LAN, а зачастую и беспроводная точка доступа LAN, и брандмауэр. Нередко они включают и функцию PAT. Что касается аппаратных средств, то у этих маршрутизаторов есть несколько портов RJ-45, помеченных как "LAN"; это порты для функции коммутатора LAN. У них есть также один порт RJ-45, помеченный как "WAN", это другой порт Ethernet, действующий как интерфейс маршрутизатора, как правило, подключаемый к модему DSL или кабельному модему, который в свою очередь подключается к Интернету (рис. 24.9).

Чтобы понять, как потребительский маршрутизатор осуществляет функции PAT изначально, следует сначала уяснить, как это делает сервер DHCP. Маршрутизатор действует как сервер DHCP на стороне LAN, используя частную сеть IP, предустановленную производителем маршрутизатора. (Изделия Cisco зачастую используют частную сеть 192.168.1.0 класса C.) Далее, на стороне WAN, маршрутизатор действу-

ет как клиент DHCP, резервируя IP-адрес у сервера DHCP провайдера ISP. Полученный от провайдера ISP адрес является не частным IP-адресом, а открытым, как показано на рис. 24.10.

Рис. 24.9. Потребительский маршрутизатор с портами LAN и WAN

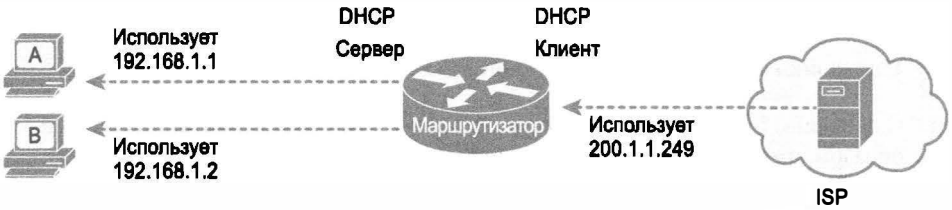


Рис. 24.10. Потребительский маршрутизатор как сервер DHCP в сети LAN и как клиент DHCP в сети WAN

ВНИМАНИЕ!

Чтобы заставить маршрутизатор корпоративного класса действовать как маршрутизатор на рис. 24.10, для порта WAN используют подкоманду интерфейса `ip address dhcp`. Эта команда указывает маршрутизатору узнать свой IP-адрес интерфейса у сервера DHCP. Конфигурация сервера DHCP использовала бы команды, подробно описанные в главе 18.

Использующий DHCP на стороне LAN и WAN, потребительский маршрутизатор обеспечивает полное соответствие IP-адресов, как при использовании PAT. У всех компьютеров в сети LAN есть частные IP-адреса, а у порта WAN — открытый IP-адрес. Все, что потребительский маршрутизатор должен сделать, — это разрешить использование PAT со стороны LAN как внутреннюю NAT и на порте WAN как внешнюю NAT (рис. 24.11).

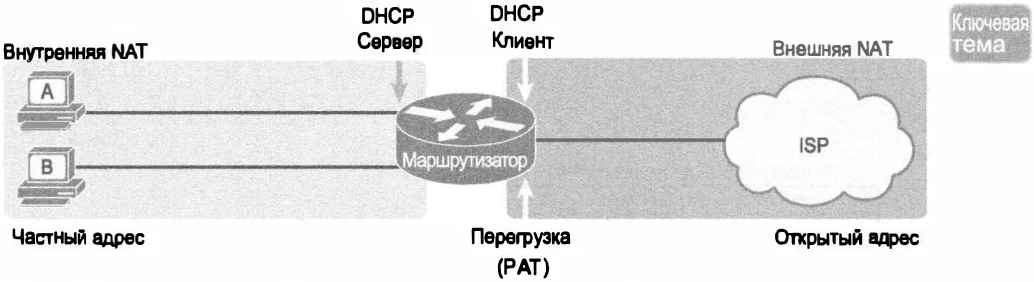


Рис. 24.11. Расположение DHCP и роли NAT/PAT на потребительском маршрутизаторе

Настройка NAT и устранение ошибок

В следующих разделах рассматривается настройка трех наиболее типичных вариантов трансляции NAT (статической, динамической и PAT), а также использование команд `show` и `debug` для устранения ошибок.

Настройка статической трансляции NAT

Настройка статической трансляции NAT по сравнению с другими ее вариантами требует наименьших действий. При этом нужно установить соответствие между локальными (частными) и глобальными (открытыми) адресами. Кроме того, необходимо указать маршрутизатору, на каких интерфейсах следует использовать трансляцию NAT, поскольку она должна быть включена не на всех интерфейсах. В частности, маршрутизатору нужно указать каждый интерфейс и является ли он внутренним или внешним. Необходимо выполнить следующие действия.

Ключевая тема

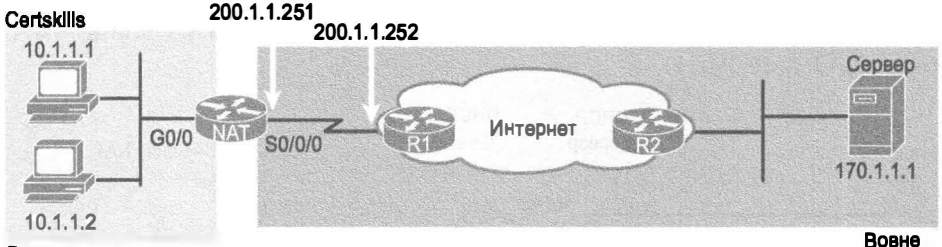
Настройка статической трансляции NAT

Этап 1 С помощью подкоманды интерфейса `ip nat inside` настроить интерфейсы таким образом, чтобы они находились во внутренней части схемы NAT

Этап 2 С помощью подкоманды интерфейса `ip nat outside` настроить интерфейсы таким образом, чтобы они находились во внешней части схемы NAT

Этап 3 Настроить статическое сопоставление с помощью команды глобального конфигурирования `ip nat inside source static` *`внутренний_локальный_адрес внутренний_глобальный_адрес`*

На рис. 24.12 показана уже знакомая вам сеть, использовавшаяся для описания статической трансляции NAT ранее в данной главе. На рисунке показано, что пользователь Certskills получил адрес 200.1.1.0 сети класса C. Вся эта сеть с маской 255.255.255.0 настроена на последовательном канале между пользователем Certskills и Интернетом. Поскольку это последовательный двухточечный канал, в данной сети используются только два из 254 действительных (возможных) IP-адресов, а 252 адреса не используются.



Конфигурация статической NAT

Внутренний локальный	Внутренний глобальный
10.1.1.1	200.1.1.1
10.1.1.2	200.1.1.2

Рис. 24.12. Смена IP-адресов NAT: частные сети

При планировании конфигурации NAT необходимо найти IP-адреса, которые будут использоваться в качестве внутренних глобальных IP-адресов. Поскольку они должны быть частью некоторого диапазона зарегистрированных IP-адресов, обычной практикой является использование дополнительных адресов в подсети, соединяющей предприятие с Интернетом, например, в данном случае — дополнительные 252 IP-адреса в сети 200.1.1.0. Маршрутизатор может быть также настроен с петлевым (loopback) интерфейсом, и ему может быть назначен IP-адрес, который является частью глобально уникального диапазона зарегистрированных IP-адресов.

В примере 24.1 приведена настройка трансляции NAT, использующая адреса 200.1.1.1 и 200.1.1.2 для двух статических сопоставлений.

Пример 24.1. Настройка статической трансляции NAT

```
NAT# show running-config
!
! Часть строк вывода опущена
!
interface Ethernet0/0
ip address 10.1.1.3 255.255.255.0
ip nat inside
!
interface Serial0/0
ip address 200.1.1.251 255.255.255.0
ip nat outside
!
ip nat inside source static 10.1.1.2 200.1.1.2
ip nat inside source static 10.1.1.1 200.1.1.1
NAT# show ip nat translations
Pro Inside global Inside local Outside local Outside global
---  200.1.1.1     10.1.1.1     ---           ---
---  200.1.1.2     10.1.1.2     ---           ---
NAT# show ip nat statistics
Total active translations: 2 (2 static, 0 dynamic; 0 extended)
Outside interfaces:
  Serial0/0
Inside interfaces:
  GigabitEthernet0/0
Hits: 100 Misses: 0
Expired translations: 0
Dynamic mappings:
```

Статические соответствия создаются с помощью команды `ip nat inside source static`. Ключевое слово `inside` означает, что NAT транслирует адреса для хостов, находящихся во внутренней части сети. Ключевое слово `source` означает, что NAT транслирует IP-адреса отправителя в пакетах, поступающих на ее внутренние интерфейсы. Ключевое слово `static` означает, что эти параметры определяют статическую запись, которая никогда не должна удаляться из таблицы NAT в связи с истечением периода времени. Поскольку в постановке задачи указано, что два хоста, 10.1.1.1 и 10.1.1.2, должны иметь доступ к Интернету, необходимы две команды `ip nat inside`.

После создания записей статической трансляции NAT маршрутизатору нужно знать, какие интерфейсы являются внутренними (inside), а какие внешними

(outside). Подкоманды интерфейса `ip nat inside` и `ip nat outside` соответствующим образом идентифицируют каждый интерфейс.

Две команды `show` выводят наиболее важную информацию о трансляции NAT. Команда `show ip nat translations` выводит две записи статической трансляции NAT, созданные в конфигурации. Команда `show ip nat statistics` выводит статистическую информацию, такую, как количество активных в данный момент записей в таблице трансляции. Эта статистика также включает в себя количество *повторных попаданий* (hit), которое увеличивается на единицу с каждым пакетом, для которого NAT должна транслировать адреса.

Настройка динамической трансляции NAT

Как можно догадаться, настройка динамической трансляции NAT в определенной степени отличается от статической, однако имеются и общие черты. Динамическая трансляция NAT по-прежнему требует идентификации каждого интерфейса как внутреннего или внешнего, и, конечно, уже не нужно задавать статическое соответствие. Для указания внутренних локальных (частных) IP-адресов, подлежащих трансляции, динамическая трансляция NAT использует *списки управления доступом* (Access Control List — ACL), а также определяет пул зарегистрированных открытых IP-адресов, которые будут выделяться для этого. Эти конкретные действия приведены ниже.

Ключевая тема **Настройка динамической трансляции NAT**

Этап 1 Как и для статической трансляции NAT, необходимо настроить интерфейсы, которые будут находиться во внутренней части проекта NAT, с помощью подкоманды интерфейса `ip nat inside`

Этап 2 Как и для статической трансляции NAT, необходимо настроить интерфейсы, которые будут находиться во внешней части проекта NAT, с помощью подкоманды интерфейса `ip nat outside`

Этап 3 Настроить список ACL, соответствующий пакетам, поступающим на внутренние интерфейсы, для которых должна быть применена трансляция NAT

Этап 4 Настроить пул открытых зарегистрированных IP-адресов с помощью команды режима глобального конфигурирования `ip nat pool` *имя первый-адрес последний-адрес* `netmask` *маска-подсети*

Этап 5 Включить динамическую трансляцию NAT, указав в команде глобального конфигурирования `ip nat inside source list` *номер-acl* `pool` *имя-пула* список ACL (этап 3) и пул (этап 4)

Следующий пример демонстрирует динамическую конфигурацию NAT, используя ту же топологию сети, что и предыдущий пример (см. рис 24.12). В данном случае в трансляции нуждаются те же два внутренних локальных адреса, 10.1.1.1 и 10.1.1.2. Но в отличие от предыдущего примера статической трансляции NAT, в примере 24.2 открытые IP-адреса (200.1.1.1 и 200.1.1.2) помещены в пул динамически присваиваемых внутренних глобальных адресов.

Пример 24.2. Настройка динамической трансляции NAT

```
NAT# show running-config
!
! Часть строк вывода опущена
!
interface GigabitEthernet0/0
ip address 10.1.1.3 255.255.255.0
ip nat inside
!
interface Serial0/0/0
ip address 200.1.1.251 255.255.255.0
ip nat outside
!
ip nat pool fred 200.1.1.1 200.1.1.2 netmask 255.255.255.252
ip nat inside source list 1 pool fred
!
access-list 1 permit 10.1.1.2
access-list 1 permit 10.1.1.1
```

Настройка динамической трансляции NAT включает в себя создание пула открытых (глобальных) адресов при помощи команды `ip nat pool`, указывающей первый и последний номера в диапазоне внутренних глобальных адресов. Например, если пул нуждался в десяти адресах, то в команде могут быть указаны номера 200.1.1.1 и 200.1.1.10, а значит, трансляция NAT сможет использовать и адрес 200.1.1.1, и 200.1.1.10.

Динамическая трансляция NAT осуществляет также проверку, используя команду `ip nat pool` с обязательным параметром `netmask`. Если диапазон адресов с учетом используемого параметра `netmask` не окажется в той же подсети, то операционная система IOS отвергнет команду `ip nat pool`. Например, при задании диапазона с младшим адресом 200.1.1.1, старшим адресом 200.1.1.2 и маской 255.255.255.252 операционная система IOS использовала бы следующие проверки и удостоверилась, что оба адреса, 200.1.1.1 и 200.1.1.2, принадлежат той же подсети.

- Адрес 200.1.1.1 с маской 255.255.255.252 подразумевает подсеть 200.1.1.0, широковещательный адрес 200.1.1.3.
- Адрес 200.1.1.2 с маской 255.255.255.252 подразумевает подсеть 200.1.1.0, широковещательный адрес 200.1.1.3.

Если бы команда имела граничные адреса 200.1.1.1 и 200.1.1.6 при той же маске 255.255.255.252, операционная система IOS отклонила бы команду. Она применила бы следующие математические вычисления и пришла бы к выводу, что номера принадлежат различным подсетям.

- Адрес 200.1.1.1 с маской 255.255.255.252 подразумевает подсеть 200.1.1.0, широковещательный адрес 200.1.1.3.
- Адрес 200.1.1.6 с маской 255.255.255.252 подразумевает подсеть 200.1.1.4, широковещательный адрес 200.1.1.7.

Между конфигурациями динамической и статической трансляции NAT в примере 24.1 есть еще одно существенное отличное, относящееся к двум параметрам

команды ip nat inside source. Версия этой команды для динамической NAT упоминает имя используемого для адресов пула NAT, в данном случае внутреннего глобального адреса Fred. Она упоминает также список ACL IP, определяющий логику распознавания для внутренних локальных IP-адресов. Таким образом, логика команды ip nat inside source list 1 pool fred в данном примере такова.

Создать записи таблицы NAT, сопоставляющие хосты, соответствующие списку ACL 1, с пакетами, поступающими на любой внутренний интерфейс, резервируя внутренние глобальные адреса из пула fred.

Проверка динамической трансляции NAT

Примеры 24.3 и 24.4 приводят доказательство того, что динамическая трансляция NAT начинается без записей в таблице NAT, но маршрутизатор реагирует на пользовательский трафик, правильно управляя функцией NAT. Пример 24.3 демонстрирует вывод команд show ip nat translations и show ip nat statistics прежде, чем пользователи создадут трафик, вынуждающий NAT выполнить некую работу. Команда show ip nat translations, выводящая записи таблицы NAT, выводит пустую строку; команда show ip nat statistics, демонстрирующая количество созданных записей таблицы NAT, отображает 0 активных трансляций.

Пример 24.3. Проверка динамической трансляции NAT перед созданием трафика

```
! Команда, показанная ниже, выводит пустую строку, поскольку
! трансляции еще не были созданы.
NAT# show ip nat translations

NAT# show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  Serial0/0
Inside interfaces:
  Ethernet0/0
Hits: 0 Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[id 1] access-list 1 pool fred refcount 0
pool fred: netmask 255.255.255.252
  start 200.1.1.1 end 200.1.1.2
  type generic, total addresses 2, allocated 0 (0%), misses 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

Команда show ip nat statistics в конце примера выводит некоторую весьма интересную информацию для поиска ошибок конфигурации в двух разных счетчиках, называемых *счетчиками пропусков* (misses), как показано в примере. При первом появлении этого счетчика в нем отображается, сколько раз появился новый пакет, требующий записи NAT, но не получивший ее. В этот момент реагирует дина-

мическая трансляция NAT и создает необходимую запись. Второй счетчик пропусков в конце вывода по команде отображает количество пропусков (misses) в пуле. Этот счетчик увеличивает свое значение на единицу, когда динамическая трансляция NAT пытается выделить новую запись в таблице NAT и не находит доступного адреса, поэтому пакет не может быть транслирован, что, вероятно, приводит к тому, что конечный пользователь не может получить доступ к приложению.

В примере 24.4 демонстрируется вывод обеих команд после установления хостом 10.1.1.1 сеанса telnet с хостом 170.1.1.1.

Пример 24.4. Проверка динамической трансляции NAT после создания трафика

```
NAT# show ip nat translations
Pro Inside global  Inside local  Outside local  Outside global
--- 200.1.1.1      10.1.1.1      ---            ---

NAT# show ip nat statistics
Total active translations: 1 (0 static, 1 dynamic; 0 extended)
Outside interfaces:
  Serial0/0
Inside interfaces:
  Ethernet0/0
Hits: 69 Misses: 1
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list 1 pool fred refcount 1
  pool fred: netmask 255.255.255.252
    start 200.1.1.1 end 200.1.1.2
    type generic, total addresses 2, allocated 1 (50%), misses 0
```

Данный пример начинается с сеанса Telnet хоста 10.1.1.1 с хостом 170.1.1.1 (на рисунке не показан), когда маршрутизатор NAT создает запись NAT в таблице. В таблице NAT отображается одна запись, сопоставляющая адрес 10.1.1.1 с адресом 200.1.1.1. Первая строка в выводе команды `show ip nat translations` отображает счетчик для 1 активной трансляции, как отмечено в таблице NAT вверху примера.

Рассмотрим последнюю выделенную строку, где команда `show ip nat statistics` отображает 1 пропуск и 69 обращений. Первый счетчик пропусков, содержащий теперь значение 1, означает, что поступил один пакет, нуждающийся в трансляции NAT, но соответствующей ему записи таблице NAT еще не было. Трансляция NAT отреагировала и добавила запись в таблицу, поэтому счетчик обращений 69 отметил следующие 69 пакетов как использующие вновь добавленную запись таблицы NAT. Второй счетчик пропусков все еще содержит значение 0, оно не увеличилось, поскольку в пуле NAT еще достаточно много доступных внутренних глобальных IP-адресов для резервирования новыми записями таблицы NAT. Обратите также внимание на то, что последняя строка выводит статистику по количеству зарезервированных членов пула (1), а также процент использования пула (50%).

По истечении определенного периода бездействия запись динамической таблицы NAT удаляется, освобождая внутренние глобальные адреса в пуле для последующего использования. Пример 24.5 демонстрирует последовательность использования двумя разными хостами внутреннего глобального адреса 200.1.1.1. Хост

10.1.1.1 использует в начале примера внутренний глобальный адрес 200.1.1.1. Затем, чтобы не ждать истечения срока давности NAT, пример очищает записи таблицы NAT при помощи команды `clear ip nat translation *`. Теперь пользователь 10.1.1.2 устанавливает сеанс telnet с хостом 170.1.1.1, и в таблице NAT появляется новая запись, использующая тот же внутренний глобальный адрес 200.1.1.1.

Пример 24.5. Пример многократного использования динамического внутреннего глобального IP-адреса

```
! Сейчас хост 10.1.1.1 использует внутренний глобальный адрес 200.1.1.1
NAT# show ip nat translations
Pro  Inside global  Inside local  Outside local  Outside global
---  200.1.1.1      10.1.1.1      ---            ---
NAT# clear ip nat translation *
!
! Затем сеанс telnet от 10.1.1.2 к 170.1.1.1; не показан
!
! Теперь хост 10.1.1.2 использует внутренний глобальный адрес 200.1.1.1
NAT# show ip nat translations
Pro  Inside global  Inside local  Outside local  Outside global
---  200.1.1.1      10.1.1.2      ---            ---
!
! Сеанс telnet от 10.1.1.1 к 170.1.1.1; не показан
!
NAT# debug ip nat
IP NAT debugging is on

Oct 20 19:23:03.263: NAT*: s=10.1.1.1->200.1.1.2, d=170.1.1.1 [348]
Oct 20 19:23:03.267: NAT*: s=170.1.1.1, d=200.1.1.1->10.1.1.1 [348]
Oct 20 19:23:03.464: NAT*: s=10.1.1.1->200.1.1.2, d=170.1.1.1 [349]
Oct 20 19:23:03.568: NAT*: s=170.1.1.1, d=200.1.1.1->10.1.1.1 [349]
```

В заключение отметим, что в конце примера 24.4 показано, что хост 10.1.1.1 связался по Telnet с другим хостом в Интернете, а также приведен вывод по команде `debug ip nat`. С помощью этой команды `debug` маршрутизатор отправляет сообщение каждый раз, когда адрес пакета транслируется для NAT. Для создания результатов вывода достаточно ввести несколько строк в сеансе Telnet с 10.1.1.1 к 170.1.1.1. Вывод отладки свидетельствует, что хост 10.1.1.1 использует теперь для этого нового соединения внутренний глобальный адрес 200.1.1.2.

Настройка перезагрузки NAT (PAT)

Вопрос настройки статической и динамической трансляции NAT важен, но настройка перегрузки NAT (PAT) имеет большее значение. Ведь это средство экономит открытые IPv4-адреса и продлевает жизнь протоколу IPv4.

Как уже отмечалось, перезагрузка NAT позволяет поддерживать много внутренних локальных адресов при наличии лишь одного или нескольких внугренних глобальных IP-адресов. По существу, транслируя частный IP-адрес и номер порта в один внутренний глобальный адрес, но с уникальным номером порта, трансляция NAT может поддерживать большое количество (более 65 тысяч) частных хостов всего лишь с одним открытым глобальным адресом.

В операционной системе IOS существуют две конфигурации PAT. Если PAT использует пул внутренних глобальных адресов, то конфигурация выглядит так же, как и у динамической трансляции NAT, за исключением того, что в конце глобальной команды `ip nat inside source list` добавляется ключевое слово `overload`. Если трансляции PAT необходим только один внутренний глобальный IP-адрес, то она может использовать один из своих IP-адресов интерфейсов. Поскольку NAT может поддерживать до 65 тысяч конкурирующих потоков данных, один открытый IP-адрес может удовлетворить потребности NAT целой организации.

Следующее определение уточняет различие между перегрузкой NAT и NAT с использованием пула.

Различия между настройкой динамической трансляции NAT и трансляции PAT, использующей пул

Ключевая тема

Выполните те же действия по конфигурированию динамической трансляции NAT, как было описано в предыдущих разделах, но включите в конце глобальной команды `nat inside source list` *ключевое слово* `overload`.

Чтобы использовать IP-адреса интерфейса в качестве единственного внутреннего глобального IP-адреса в трансляции NAT с перезагрузкой, следует выполнить описанные ниже действия.

Настройка PAT с использованием IP-адреса интерфейса

Ключевая тема

Этап 1 Как и в случае с динамической и статической трансляцией NAT, необходимо настроить внутренние интерфейсы с помощью подкоманды интерфейса `ip nat inside`

Этап 2 Как и в случае с динамической и статической трансляцией NAT, необходимо настроить внешние интерфейсы с помощью подкоманды интерфейса `ip nat outside`

Этап 3 Как и в случае с динамической трансляцией NAT, необходимо настроить список доступа ACL, которому должны соответствовать пакеты, поступающие на внутренние интерфейсы

Этап 4 Задайте в конфигурации команду глобального конфигурирования `ip nat inside source list` *`номер-acl`* `interface` *`тип/номер`* `overload`, которая ссылается на список ACL, созданный на этапе 3, и на интерфейс, IP-адрес которого будет использоваться для трансляции адресов

В примере 24.2 приведена настройка динамической трансляции NAT. Для ее преобразования в конфигурацию PAT вместо команды `ip nat inside source list 1 pool fred overload` следует ввести такую же команду, добавив ключевое слово `overload`.

В следующем примере показана настройка PAT с использованием IP-адреса одного интерфейса. На рис. 24.13 показана та же сеть с некоторыми изменениями. В данном случае провайдер ISP предоставил пользователю Certskills подмножество сети 200.1.1.0: подсеть CIDR 200.1.1.248/30. Иными словами, эта подсеть имеет два используемых адреса: 200.1.1.249 и 200.1.1.250. Эти адреса используются на концах последовательного канала между пользователем Certskills и его провайдером ISP. Функция NAT на маршрутизаторе Certskills транслирует все NAT-адреса в последовательный IP-адрес 200.1.1.249.

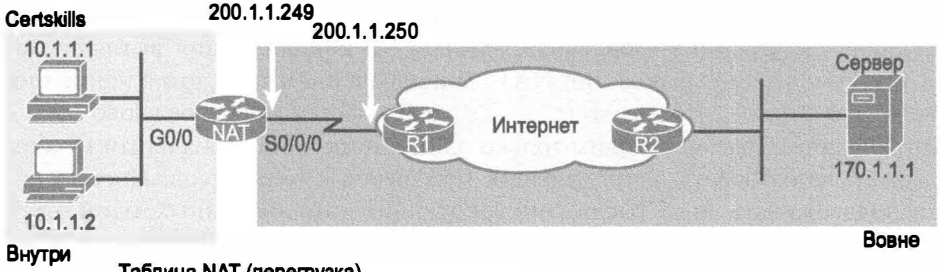


Таблица NAT (перегрузка)

Внутренний локальный	Внутренний глобальный
10.1.1.1: 3212	200.1.1.249: 3212
10.1.1.2: 3213	200.1.1.249: 3213
10.1.1.2: 38913	200.1.1.249: 38913

Рис. 24.13. Перезагрузка NAT и трансляция PAT

В примере 24.6, иллюстрирующем конфигурацию перезагрузки NAT, выполняется трансляция с использованием только глобального адреса 200.1.1.249, поэтому пул NAT не нужен. В данном примере, как предполагается на рис. 24.13, хост 10.1.1.1 создает два соединения Telnet, а хост 10.1.1.2 — одно, что приводит к созданию трех динамических записей NAT, каждая из которых использует внутренний глобальный адрес 200.1.1.249, но имеет уникальный номер порта.

Пример 24.6. Настройка перезагрузки NAT

```
NAT# show running-config
!
! Часть строк вывода опущена
!
interface GigabitEthernet0/0
  ip address 10.1.1.3 255.255.255.0
  ip nat inside
!
interface Serial0/0/0
ip address 200.1.1.249 255.255.255.252
ip nat outside
!
ip nat inside source list 1 interface Serial0/0/0 overload
!
access-list 1 permit 10.1.1.2
access-list 1 permit 10.1.1.1
!

NAT# show ip nat translations
Pro  Inside global     Inside local   Outside local   Outside global
tcp  200.1.1.249:3212  10.1.1.1:3212  170.1.1.1:23    170.1.1.1:23
tcp  200.1.1.249:3213  10.1.1.2:3213  170.1.1.1:23    170.1.1.1:23
tcp  200.1.1.249:38913 10.1.1.2:38913 170.1.1.1:23    170.1.1.1:23
NAT# show ip nat statistics
Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Outside interfaces:
  Serial0/0/0
Inside interfaces:
  Serial0/0/0
```

```
Hits: 103 Misses: 3
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list 1 interface Serial0/0/0 refcount 3
```

Команда `ip nat inside source list 1 interface serial 0/0/0 overload` имеет несколько параметров, однако если читателю понятна конфигурация динамической трансляции NAT, то понять назначение новых параметров будет несложно. Параметр `list 1` имеет в данном случае тот же смысл, что и для динамической трансляции NAT: транслируются внутренние локальные IP-адреса, соответствующие условиям списка ACL 1. Параметр `interface serial 0/0/0` означает, что единственным доступным внутренним глобальным адресом является номер последовательного интерфейса 0/0/0 маршрутизатора NAT. Последний параметр `overload` означает, что включена перезагрузка. Без этого параметра маршрутизатор не будет выполнять перезагрузку, как это происходит в случае динамической трансляции NAT.

Как видно из вывода по команде `show ip nat translations`, в таблицу NAT были добавлены три трансляции. До выполнения этой команды хост 10.1.1.1 создает одно соединение Telnet с 170.1.1.1, а хост 10.1.1.2 — два. Маршрутизатор создает по одной записи таблицы NAT для каждой уникальной комбинации внутреннего локального IP-адреса и порта.

Устранение ошибок в конфигурации NAT

Большинство вопросов устранения ошибок NAT связано с необходимостью сделать конфигурацию корректной (правильной). Ниже обобщены некоторые советы, относящиеся к наиболее распространенным проблемам настройки NAT. Затем объясняется одна общая проблема маршрутизации, которая не позволяет работать трансляции NAT и связана главным образом с обеспечением правильности конфигурации.

Наиболее типичные ошибки при настройке трансляции NAT

- Проверьте, содержит ли конфигурация команды интерфейса `ip nat inside` или `ip nat outside`. Эти команды включают на интерфейсах трансляцию NAT; важно не перепутать получателя (внутренний/внешний).
- Для статической трансляции NAT проверьте, выводит ли команда `ip nat inside source static` сначала внутренний локальный адрес, а затем внутренний глобальный IP-адрес.
- Для динамической трансляции NAT проверьте, настроен ли список ACL на проверку того, что пакеты, отправленные внутренним хостом, соответствуют пакетам этого хоста, до того, как началась трансляция NAT. Например, если внутренний локальный хост имеет адрес 10.1.1.1 и должен транслироваться в адрес 200.1.1.1, то следует проверить, соответствует ли список ACL адресу отправителя 10.1.1.1, а не 200.1.1.1.
- Для динамической трансляции NAT без PAT убедитесь в том, что пул имеет достаточно IP-адресов. Одним из признаков недостаточного количества ад-

ресов может служить растущее значение второго счетчика пропусков в выводе команды `show ip nat statistics`, а также просмотр всего пула адресов, уже находящегося в таблице NAT.

- Для трансляции PAT легко забыть, что в команде `ip nat inside source list` *должно быть* добавлено ключевое слово `overload`. Без него трансляция NAT работает, а трансляция PAT — нет, и это часто приводит к тому, что пакеты пользователя не транслируются и хосты не могут получить доступ к Интернету.
- Может оказаться, что трансляция NAT была настроена корректно, однако на одном из интерфейсов имеется список ACL, который отбрасывает пакеты. Отметим, что операционная система IOS обрабатывает списки ACL до трансляции NAT для пакетов, поступающих на интерфейс, и после трансляции адресов для пакетов, покидающих интерфейс.
- Удостоверьтесь, что некий пользовательский трафик поступает на внутренний интерфейс маршрутизатора NAT, вынуждая его осуществлять трансляцию. Трансляция NAT реагирует на пакеты, поступающие на интерфейс, а затем использует логику, заданную в конфигурации NAT. Конфигурация NAT может быть правильной, но если не поступает никакого входящего трафика, соответствующего конфигурации NAT, она ничего не делает.

В заключение отметим, что на работу функции NAT маршрутизатора могут оказывать влияние проблемы, существующие на другом маршрутизаторе. Маршрутизаторы во внешней части сети, зачастую в Интернете, должны быть способны маршрутизировать пакеты на внутренние глобальные адреса, настроенные на маршрутизаторе NAT. Например, на рис. 24.4 показан поток пакетов из внутренней части сети во внешнюю и обратно. Рассматривая поток из внешней части сети во внутреннюю, отметим, что маршрутизаторы Интернета должны знать, как следует маршрутизировать пакеты на открытый зарегистрированный IP-адрес 200.1.1.1. Обычно этот диапазон адресов анонсируется протоколом динамической маршрутизации. Поэтому, если конфигурация выглядит правильной, следует проверить маршруты как на маршрутизаторе NAT, так и на других маршрутизаторах и убедиться, что они могут пересылать пакеты на основе адресов, используемых на обеих сторонах маршрутизатора, выполняющего функцию NAT.

Обзор

Резюме

- Маршрутизация CIDR представляет собой глобальное соглашение о назначении адресов, которое определяет, каким образом Агентство по назначению адресов Интернет (IANA), его филиалы и провайдеры ISP назначают глобально уникальные адресные пространства протокола IPv4 отдельным организациям.
- У бесклассовой междоменной маршрутизации, определенной в документе RFC 4632, есть две основных задачи.
  - Во-первых, бесклассовая междоменная маршрутизация определяет способ присвоения открытых IP-адресов во всем мире, чтобы обеспечить объединение или суммирование маршрутов.
  - Во-вторых, бесклассовая междоменная маршрутизация определяет правила, позволяющие провайдерам ISP присваивать открытые IP-адреса и кроме как блоками во всю сеть класса A, B или C. Бесклассовая междоменная маршрутизация позволяет провайдерам услуг Интернета присвоить блок открытых IPv4-адресов такого размера, который лучше удовлетворяет потребности конкретного клиента.
- Бесклассовая междоменная маршрутизация требует использования бесклассового протокола маршрутизации, который, по определению, посылает наряду с каждым маршрутом маску.
- При создании частной сети, не имеющей соединения с Интернетом, можно использовать сетевые IP-адреса, называемые *частными интерсетями*, как это определено в документе RFC 1918 "Выделение адресов для частных интерсетей". В этом документе RFC определено множество сетей, которые никогда не будут назначены какой-либо организации в качестве зарегистрированного сетевого номера. Вместо чьих-либо зарегистрированных сетевых адресов можно использовать номера из диапазона, которые никем не используются в открытой сети Интернет.
- Трансляция NAT, определенная в документе RFC 3022, позволяет хосту, не имеющему действительного, зарегистрированного глобально уникального IP-адреса, осуществлять связь с другими хостами через Интернет.
- Трансляция NAT обеспечивает это за счет использования действительных зарегистрированных IP-адресов для представления данного частного адреса всем остальным хостам Интернета. Трансляция NAT заменяет частные IP-адреса открытыми зарегистрированными IP-адресами в каждом пакете протокола IP.
- В терминологии Cisco корпоративная сеть, использующая частные адреса, а следовательно, требующая использования NAT, является "внутренней" частью сети. Подключенный к Интернету интерфейс трансляции NAT является "внешней" частью сети. Хост, которому необходима трансляция NAT, имеет

IP-адрес, который он использует внутри сети, и ему требуется IP-адрес, который будет представлять его вне этой сети. Поскольку хосту фактически нужны два разных адреса для его представления, требуются два термина. В документации Cisco частные IP-адреса, используемые во внутренней сети, называются *внутренними локальными адресами*, а адреса, используемые для представления хоста в Интернете, — *внутренними глобальными адресами*.

- В сравнении со статической динамическая трансляция NAT имеет как сходства, так и отличия. Как и в случае использования статической трансляции NAT, маршрутизатор NAT устанавливает взаимно однозначное соответствие между внутренним локальным и внутренним глобальным адресами и изменяет IP-адреса в пакетах, когда они входят во внутреннюю сеть и выходят из нее. Однако преобразование внутренних локальных адресов во внутренние глобальные адреса происходит динамически.
- Динамическая трансляция NAT создает пул возможных внутренних глобальных адресов и определяет критерий соответствия для определения того, какие внутренние глобальные IP-адреса должны транслироваться с помощью NAT.
- Функция перезагрузки NAT, называемая также трансляцией адресов портов (PAT), решает эту проблему. Перезагрузка позволяет трансляции NAT выполнить масштабирование для поддержки многих клиентов с использованием всего лишь нескольких открытых IP-адресов.
- Поскольку поле номера порта состоит из 16 бит, перезагрузка NAT позволяет использовать более 65 тысяч номеров портов, что позволяет ей выполнять масштабирование без необходимости иметь много зарегистрированных IP-адресов.
- Настройка статической трансляции NAT по сравнению с другими ее вариантами требует наименьших действий. При этом нужно установить соответствие между локальными (частными) и глобальными (открытыми) адресами. Кроме того, необходимо указать маршрутизатору, на каких интерфейсах следует использовать трансляцию NAT, поскольку она должна быть включена не на всех интерфейсах. В частности, маршрутизатору нужно указать каждый интерфейс и является ли он внутренним или внешним. Необходимо выполнить следующие действия.

  Этап 1 С помощью подкоманды интерфейса `ip nat inside` настроить интерфейсы таким образом, чтобы они находились во внутренней части схемы NAT

  Этап 2 С помощью подкоманды интерфейса `ip nat outside` настроить интерфейсы таким образом, чтобы они находились во внешней части схемы NAT

  Этап 3 Настроить статическое сопоставление с помощью команды глобального конфигурирования `ip nat inside source static` *`внутренний_локальный_адрес внутренний_глобальный_адрес`*

- Как можно догадаться, настройка динамической трансляции NAT в определенной степени отличается от статической, однако имеются и общие черты. Динамическая трансляция NAT по-прежнему требует идентификации каждого интерфейса как внутреннего или внешнего, и, конечно, уже не нужно задавать статическое соответствие. Для указания внутренних локальных

(частных) IP-адресов, подлежащих трансляции, динамическая трансляция NAT использует списки управления доступом (ACL), а также определяет пул зарегистрированных открытых IP-адресов, которые будут выделяться для этого. Эти конкретные действия приведены ниже.

Этап 1 Как и для статической трансляции NAT, необходимо настроить интерфейсы, которые будут находиться во внутренней части проекта NAT, с помощью подкоманды интерфейса `ip nat inside`

Этап 2 Как и для статической трансляции NAT, необходимо настроить интерфейсы, которые будут находиться во внешней части проекта NAT, с помощью подкоманды интерфейса `ip nat outside`

Этап 3 Настроить список ACL, соответствующий пакетам, поступающим на внутренние интерфейсы, для которых должна быть применена трансляция NAT

Этап 4 Настроить пул открытых зарегистрированных IP-адресов с помощью команды режима глобального конфигурирования `ip nat pool` *`имя первый-адрес последний-адрес`* `netmask` *`маска-подсети`*

Этап 5 Включить динамическую трансляцию NAT, указав в команде глобального конфигурирования `ip nat inside source list` *`номер-acl`* `pool` *`имя-пула`* список ACL (этап 3) и пул (этап 4)

- Следующее определение уточняет различие между перегрузкой NAT и NAT с использованием пула.
- ***Выполните те же действия по конфигурированию динамической трансляции NAT, как было описано в предыдущих разделах, но включите в конце глобальной команды*** `ip nat inside source list` ***ключевое слово*** `overload`.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Как расшифровывается аббревиатура CIDR?

   А) Классовая маршрутизация по умолчанию (Classful IP Default Routing).

   Б) Классовая маршрутизация D-класса (Classful IP D-class Routing).

   В) Классовая междоменная маршрутизация (Classful Interdomain Routing).

   Г) Бесклассовая маршрутизация по умолчанию (Classless IP Default Routing).

   Д) Бесклассовая маршрутизация D-класса (Classless IP D-class Routing).

   Е) Бесклассовая междоменная маршрутизация (Classless Interdomain Routing).

2. Какая из приведенных ниже суммарных подсетей представляет маршруты, которые могли бы быть созданы для уменьшения размера таблиц маршрутизации Интернета?

   А) 10.0.0.0 255.255.255.0.

   Б) 10.1.0.0 255.255.0.0.

   В) 200.1.1.0 255.255.255.0.

   Г) 200.1.0.0 255.255.0.0.

3. Какие из приведенных ниже адресов согласно документу RFC 1918 не являются частными? (Выберите два ответа.)

   А) 172.31.1.1.

   Б) 172.33.1.1.

   В) 10.255.1.1.

   Г) 10.1.255.1.

   Д) 191.168.1.1.

4. При использовании статической трансляции NAT, осуществляющей трансляцию только внутренних адресов, что приводит к созданию записей таблицы NAT?

   А) Первый пакет, пересылаемый из внутренней сети во внешнюю.

   Б) Первый пакет, пересылаемый из внешней сети во внутреннюю.

   В) Конфигурация, использующая команду `ip nat inside source`.

   Г) Конфигурация, использующая команду `ip nat outside source`.

5. При использовании динамической трансляции NAT, осуществляющей трансляцию только внутренних адресов, что приводит к созданию записей таблицы NAT?

   А) Первый пакет, пересылаемый из внутренней сети во внешнюю.

   Б) Первый пакет, пересылаемый из внешней сети во внутреннюю.

   В) Конфигурация, использующая команду `ip nat inside source`.

   Г) Конфигурация, использующая команду `ip nat outside source`.

6. Трансляция NAT была настроена для трансляции адресов отправителя пакетов, полученных из внутренней части сети, однако лишь для некоторых хостов, как определено списком управления доступом. Какая из приведенных ниже команд однозначно идентифицирует хосты?

   А) `ip nat inside source list 1 pool barney.`

   Б) `ip nat pool barney 200.1.1.1 200.1.1.254 netmask 255.255.255.0.`

   В) `ip nat inside.`

   Г) `ip nat inside 200.1.1.1 200.1.1.2.`

7. Рассмотрим следующие команды конфигурирования.

```
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
interface Serial0/0
 ip address 200.1.1.249 255.255.255.252
ip nat inside source list 1 interface Serial0/0
access-list 1 permit 10.1.1.0 0.0.0.255
```

   Если эта конфигурация предназначена для включения перезагрузки NAT отправителя, то какая из приведенных ниже команд может оказаться полезной для завершения настройки? (Выберите два ответа.)

   А) Команда `ip nat outside`.

   Б) Команда `ip nat pat`.

   В) Ключевое слово `overload`.

   Г) Команда `ip nat pool`.

8. Рассмотрим приведенный ниже вывод команды `show` в маршрутизаторе, который настроен для выполнения динамической трансляции NAT.

```
-- Inside Source
access-list 1 pool fred refcount 2288
pool fred: netmask 255.255.255.240
  start 200.1.1.1 end 200.1.1.7
  type generic, total addresses 7, allocated 7 (100%), misses 965
```

Пользователи жалуются на невозможность зайти в Интернет. Какова наиболее вероятная причина этого?

А) Судя по информации вывода, проблема не связана с использованием трансляции NAT.

Б) Пул трансляции NAT не имеет достаточно записей для выполнения всех запросов.

В) Стандартный список управления доступом ACL 1 использовать невозможно; следует использовать расширенный список управления доступом ACL.

Г) Вывод по данной команде не предоставляет достаточной информации для выяснения причины возникшей проблемы.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 24.3.

Таблица 24.3. Ключевые темы главы 24

Элемент	Описание	Страница
Рис. 24.1	Типичный случай использования маршрутизации CIDR	702
Табл. 24.1	Пространство частных адресов согласно документу RFC 1918	703
Рис. 24.2	Замена IP-адресов при использовании трансляции NAT: частная адресация	704
Рис. 24.4	Диаграмма типичной сети NAT и ключевые термины трансляции NAT	706
Табл. 24.2	Термины трансляции NAT и их определения	706
Рис. 24.8	Принципы сохранения адресов при перезагрузке NAT (PAT)	710
Рис. 24.11	Расположение DHCP и роли NAT/PAT на потребительском маршрутизаторе	711
Список	Настройка статической трансляции NAT	712
Список	Настройка динамической трансляции NAT	714
Предложение	Различия между настройкой динамической трансляции NAT и трансляции PAT, использующей пул	719
Список	Настройка PAT с использованием IP-адреса интерфейса	719
Список	Наиболее типичные ошибки при настройке трансляции NAT	721

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

CIDR, внутренний глобальный адрес (inside global), внутренний локальный адрес (inside local), перегрузка NAT (NAT overload), внешний глобальный адрес (outside global), внешний локальный адрес (outside local), PAT, частная сеть IP (private IP network)

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 24.4 приведен список команд конфигурации, а в табл. 24.5 пользовательские команды главы. Фактически команды стоит запомнить, чтобы лучше понять материал главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 24.4. Команды конфигурации главы 24

Команда	Описание
`ip nat {inside | outside}`	Команда режима настройки интерфейса для включения трансляции NAT и указания, где находится интерфейс — во внутренней или во внешней части сети
`ip nat inside source {list {`*`номер_списка | имя_списка`*`}} {interface `*`тип номер`*` | pool `*`имяпула`*`} [overload]`	Команда режима глобального конфигурирования, включающая трансляцию NAT, со ссылкой на список доступа ACL, который определяет, какой из отправителей обращается к NAT, а также указывает интерфейс или пул, в котором следует искать глобальные адреса
`ip nat pool `*`имя начальный_ip конечный_ip`*` {netmask `*`маска_сети`*` | prefix-length `*`длина_префикса`*`}`	Глобальная команда для указания пула адресов трансляции NAT
`ip nat source static `*`внутренний_ip {внешний_ip | идентификатор_интерфейса}`*	Глобальная команда, перечисляющая соответствующие внутренние и внешние адреса (или внешние интерфейсы, IP-адреса которых должны использоваться), подлежащие добавлению к таблице трансляции NAT

Таблица 24.5. Пользовательские команды главы 24

Команда	Описание
`show ip nat statistics`	Выводит список счетчиков пакетов и записи таблицы NAT, а также базовую информацию конфигурирования
`show ip nat translations [verbose]`	Отображает таблицу NAT
`clear ip nat translation {* | [inside` *глобальный_ip локальный_ip*`] [outside` *локальный_ip глобальный_ip*`]}`	Очищает все или некоторые динамические записи таблицы NAT в зависимости от используемых параметров
`clear ip nat translation` *протокол* `inside` *глобальный_ip глобальный_порт локальный_ip локальный_порт* `[outside` *локальный_ip глобальный_ip*`]`	Очищает некоторые из динамических записей в таблице NAT, в зависимости от использованных параметров
`debug ip nat`	Создает сообщения системного журнала, в котором описаны все пакеты, адреса которых были транслированы с помощью NAT

Ответы на контрольные вопросы:

1 Е. **2** Г. **3** Б и Д. **4** В. **5** А. **6** А. **7** А и В. **8** Б.

Обзор части VI

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части VI

Задание	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей команд по категориям		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав только этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения к книге.

Ответы на вопросы

Ответьте на вопросы обзора этой части, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров только этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения к книге.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если понятны не все их подробности, уделите время повторному изучению.

Создайте диаграмму связей команд по категориям

Подобно частям II и IV этой книги, в части VI представлено множество новых команд CLI. Такое большое количество команд трудно запомнить, поэтому имеет смысл выполнить специальные упражнения, позволяющие лучше понять подробности и вспомнить их при необходимости.

Задача этого упражнения заключается в том, чтобы помочь запомнить команды. Оно не сосредоточивается на деталях и каждом отдельном параметре каждой команды или даже их значении. Цель в том, чтобы вспомнить их, когда они встретятся в реальности или на экзамене.

Создайте диаграмму связей со следующими категориями команд.

Нумерованный стандартный список ACL IPv4, нумерованный расширенный список ACL IPv4, именованный список ACL IPv4, защита маршрутизатора и коммутатора, NAT и остальное.

В этой диаграмме связей для каждой категории вспомните все команды конфигурации и все пользовательские команды (главным образом команды `show`). По каждой категории сгруппируйте команды конфигурации отдельно от пользовательских команд. Пример диаграммы связей приведен на рис. Ч6.1.

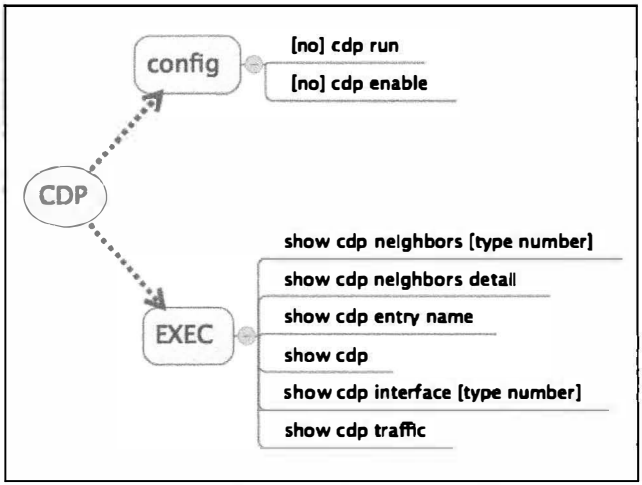


Рис. Ч6.1. Пример диаграммы связей

ВНИМАНИЕ!

Более подробная информация по этой теме приведена в разделе "О диаграммах связей" введения к данной книге.

И наконец, работая над этим проектом, имейте в виду следующее.

- Выполните это задание, не подглядывая в книгу или свои записи.
- Завершив упражнение, сверьте результат с таблицами команд в концы глав и обратите внимание на первоначально забытые команды.
- Не волнуйтесь особенно о каждом пропущенном параметре или точности синтаксиса — достаточно записать лишь несколько первых слов команды.
- Делайте для последующего анализа примечания об абсолютно понятных командах и командах, в которых вы меньше уверены.
- Повторите это упражнение впоследствии, когда появится десять свободных минут, и сравните результат с примечаниями, сделанными в прошлый раз.

Ответы приведены в приложении П (Appendix O) на веб-сайте, но ваши диаграммы связей могут выглядеть иначе.

Диаграммы связей обзора части VI

Диаграмма	Описание	Где сохранен результат
1	Диаграмма связей команд	

До сих пор в этой книге протокол IP версии 6 (IPv6) по большей части игнорировался. Данная часть, напротив, объединяет в пять глав все темы, специфические для протокола IPv6.

В главах этой части нередко повторяются темы, обсуждавшиеся ранее для протокола IPv4, но теперь в сравнении с ним. Конечно, многие детали протоколов IPv4 и IPv6 отличаются, но многие базовые концепции IP-адресации, создания подсетей, перенаправления и применения протоколов маршрутизации остаются теми же. Главы данной части рассматривают эти основополагающие концепции, а также специфические подробности перенаправления пакетов IPv6 с одного хоста на другой.

Часть VII. Протокол IP версии 6

ГЛАВА 25

Основы протокола IP версии 6

Протокол IPv4 был основной и весьма полезной частью протокола TCP/IP и Интернета. На протяжении почти всей истории Интернета в большинстве корпоративных сетей, использующих протокол TCP/IP, протокол IPv4 был базовым протоколом, определяющим адресацию и маршрутизацию. Но даже при том, что у протокола IPv4 есть множество великолепных качеств, у него на самом деле есть несколько недостатков, из-за которых возникла необходимость в создании протокола на замену: IP версии 6 (IPv6).

Протокол IPv6 определяет те же общие функции, что и протокол IPv4, но с иными методами реализации этих функций. Например, оба протокола, IPv4 и IPv6, определяют адресацию, концепции создания подсетей (разделения больших групп адресов на меньшие группы), заголовки IPv4 и IPv6 пакетов, а также правила перенаправления пакетов. Однако детали протокол IPv6 реализует по-другому, например, использует 128-битовые IPv6-адреса, а не 32-битовые, как протокол IPv4.

В этой главе основное внимание уделяется базовым функциям сетевого уровня адресации и маршрутизации. В первом разделе рассматриваются в основном концепции, а второй раздел посвящен специфическим особенностям написания и ввода IPv6-адресов.

В этой главе рассматриваются следующие экзаменационные темы

Работа сетей передачи данных IP

Передача данных между двумя хостами по сети.

IP-адресация (IPv4/IPv6)

Выбор подходящей схемы IPv6-адресации, удовлетворяющей требованиям адресации в среде LAN/WAN.

Описание IPv6-адреса.

Глобальный одноадресатный.

Технологии маршрутизации IP

Различия методов маршрутизации и протоколов маршрутизации:

Ближайшая точка перехода.

Таблица IP-маршрутизации.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Введение в IPv6

Протокол IP версии 6 (IPv6) предназначен для замены протокола IP версии 4 (IPv4).

К сожалению, это утверждение создает больше вопросов, чем ответов. Зачем заменять протокол IPv4? Если его нужно заменять, то как быстро это произойдет? Что будет, когда компании и Интернет заменят протокол IPv4 на IPv6? Перечень вопросов продолжается.

Хотя в этой вводной главе невозможно рассмотреть все причины замены протокола IPv4 протоколом IPv6, наиболее очевидной из них является стремительный рост сетей TCP/IP. Протокол IPv4 использует 32-битовые адреса, общее допустимое количество которых составляет несколько миллиардов. Но этого, казалось бы, огромного количества оказалось недостаточно. Протокол IPv6 использует 128-битовые адреса и потенциально позволяет создать их более 10 000 000 000 000 000 000 000 000 000.

Тот факт, что протокол IPv6 использует иной размер поля адреса и несколько иные правила адресации, означает также изменение многих других протоколов и функций. Например, маршрутизация IPv4 (процесс перенаправления пакетов) полагается на понимание IPv4-адресов. Для обеспечения маршрутизации IPv6 маршрутизаторы должны понимать IPv6-адреса и маршруты. Чтобы динамически изучать маршруты к подсетям IPv6, протоколы маршрутизации должны поддерживать также и правила адресации IPv6, включая правила создания подсетей IPv6. В результате переход с протокола IPv4 на IPv6 потребует изменения не только протокола IP, но и многих других протоколов.

В первом разделе этой главы обсуждаются некоторые из причин перехода с протокола IPv4 на IPv6, а также протоколы, которые должны измениться в результате.

Исторические причины перехода на протокол IPv6

За прошедшие сорок лет Интернет прошел путь от младенчества до всемирного явления. Интернет появился в конце 1960-х годов в результате университетских исследований и развивался в 1970-е годы как сеть ARPANET. Его быстрый рост пришелся на 1980-е годы, прежде всего благодаря исследовательским организациям и университетам, подключившимся к его разработке. К началу 1990-х Интернет начал использоваться в коммерческих целях, позволяя предоставлять услуги и вести торговлю, что обусловило еще более быстрый его рост. Некоторые из этих основных вех представлены на рис. 25.1.

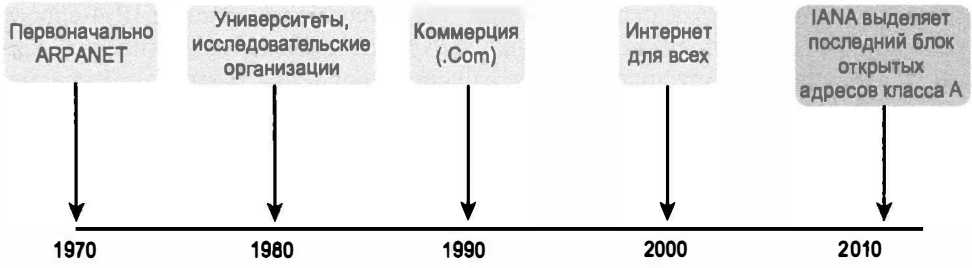


Рис. 25.1. Некоторые из основных вех развития Интернета

Обратите внимание, что эволюция на рисунке завершается в феврале 2011 событием присвоения группой IANA/ICANN последнего блока открытых IPv4-адресов сети класса A. Это было очень важным событием для Интернета, приблизившим тот день, когда очередная компания просто не сможет получить новый блок открытых IPv4-адресов.

Другими словами, однажды новая компания захочет подключиться к Интернету, но не сможет, поскольку протокол IPv4 не имеет больше открытых адресов.

Хотя пресса сделала грандиозное заявление об исчерпывании IPv4-адресов только в 2011 году, специалисты Интернета знали об этой потенциальной проблеме с конца 1980-х. Сама проблема, названная в общем *проблемой исчерпания IPv4-адресов*, возможно, также вызванная существенным ростом Интернета в 1990-х годах, грозила остановкой его роста! Что-то нужно было делать.

Надеясь отсрочить день, когда будут исчерпаны открытые IPv4-адреса, группа IETF выработала несколько краткосрочных решений, чтобы продлить жизнь протокола IPv4 подольше. Двумя основными краткосрочными решениями были *трансляция сетевых адресов и трансляция адресов с использованием портов* (Network Address Translation/Port Address Translation — NAT/PAT), а также *бесклассовая междоменная маршрутизация* (Classless Interdomain Routing — CIDR). Оба решения сработали прекрасно. Хотя сообщество Интернета надеялось продлить жизнь протокола IPv4 хотя бы еще на несколько лет, практически оно продлило ее на несколько десятилетий (рис. 25.2).

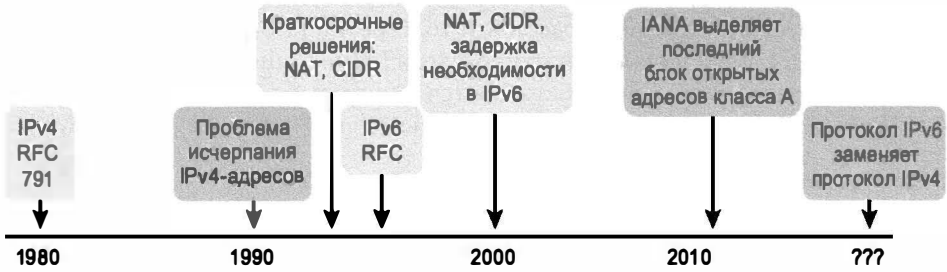


Рис. 25.2. Процесс исчерпания IPv4-адресов, а также краткосрочные и долгосрочные решения

ВНИМАНИЕ!

На сайте `www.potaroo.net` приведено много интересных статистических данных о росте Интернета, включая исчерпание IPv4-адресов.

Несмотря на то что краткосрочные решения проблемы исчерпания IPv4-адресов и позволили нам всем использовать протокол IPv4 еще несколько десятилетий, протокол IPv6 предоставит миру долгосрочное решение проблемы. Протокол IPv6 заменит протокол IPv4 как базовый протокол третьего уровня, с новым заголовком IPv6 и новыми IPv6-адресами. Размер адреса обеспечивает огромное количество адресов, решая проблему нехватки адресов на многие поколения вперед (мы надеемся).

Далее мы рассмотрим протокол IPv6 в сравнении с протоколом IPv4, сосредоточившись на их общих чертах. В частности, сравним протоколы (включая адреса), маршрутизацию, протоколы маршрутизации и многие другие сопутствующие аспекты.

ВНИМАНИЕ!

Может возникнуть вопрос: почему следующая версия протокола IP не называется IP версии 5? Ранее было предпринято усилие по созданию новой версии протокола IP, и ей был присвоен номер версии 5. Протокол IPv5 не удовлетворял текущему рабочему этапу разработки стандартов. Но чтобы снять любые проблемы, поскольку версия 5 уже упоминалась в некоторых документах, следующая модификация протокола IP получила номер 6.

Протоколы IPv6

Главная цель базового протокола IPv6 та же, что и у протокола IPv4. Базовый протокол IPv6 определен в документе RFC 2460 как концепция пакета, адреса этих пакетов, а также роли хостов и маршрутизаторов. Установленные правила позволяют устройствам перенаправлять пакеты хостам через несколько маршрутизаторов так, чтобы они достигли правильного хоста получателя. (Те же концепции для протокола IPv4 определяет документ RFC 791.)

- **Прежняя версия 2 протокола OSPF обновлена до версии 3.** Протокол OSPF версии 2 годился для протокола IPv4, но не для IPv6, поэтому для его поддержки была создана более новая версия, OSPF версии 3.
- **Протокол ICMP обновлен до версии 6.** Протокол управляющих сообщений Интернета (ICMP) годился для протокола IPv4, но для поддержки протокола IPv6 его пришлось изменить. Новое имя протокола — ICMPv6.
- **Протокол ARP заменен протоколом обнаружения соседних устройств.** Для протокола IPv4 используемые соседями MAC-адреса обнаруживает протокол преобразования адресов (ARP). Протокол IPv6 заменяет протокол ARP более общим *протоколом обнаружения соседних устройств* (Neighbor Discovery Protocol — NDP).

ВНИМАНИЕ!

Если посетить некий веб-сайт, перечисляющий запросы на комментарии, например `www.rfc-editor.org`, то можно найти почти 300 запросов на комментарии, в заголовке которых упоминается протокол IPv6.

Хотя термин IPv6 используется довольно широко и включает множество протоколов, новый 128-битовый IPv6-адрес определяет один конкретный протокол под названием IPv6. Конечно, запись этих адресов в двоичном виде была бы проблемой — они, вероятно, даже не поместятся по ширине на листе бумаги! Протокол IPv6 определяет более короткий, шестнадцатеричный формат, требующий максимум 32 шестнадцатеричных цифры (одна шестнадцатеричная цифра на 4 бита), а также методы сокращения и шестнадцатеричных адресов.

Например, ниже перечислены IPv6-адреса, каждый из не более чем 32 шестнадцатеричных цифр.

```
2345:1111:2222:3333:4444:5555:6666:AAAA
2000:1:2:3:4:5:6:A
FE80::1
```

В следующем разделе, "Адресация IPv6, формат и соглашения", рассматриваются конкретные особенности представления IPv6-адресов, включая правила сокращения шестнадцатеричных значений адреса.

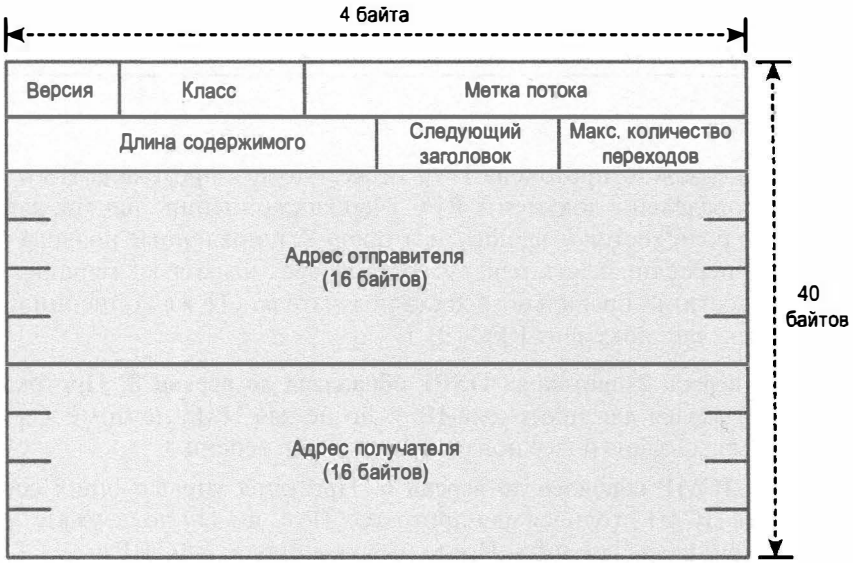


Рис. 25.3. Заголовок IPv6

Маршрутизация IPv6

Подобно многим другим функциям, маршрутизация IPv6 выглядит точно так же, как и маршрутизация IPv4, но с различиями в специфических подробностях. Сначала обсудим концепции, общие для протоколов IPv6 и IPv4.

Ключевая тема

Подобия протоколов IPv4 и IPv6

- Чтобы интерфейс устройства конечного пользователя мог создавать и передавать пакеты IPv6, он нуждается в IPv6-адресе.
- Хостам конечного пользователя должен быть известен IPv6-адрес стандартного маршрутизатора, на который хост посылает пакеты IPv6, если хост получателя находится в другой подсети.
- При передаче пакета маршрутизаторы IPv6 извлекают и повторно инкапсулируют каждый пакет IPv6.
- Маршрутизаторы IPv6 принимают решение о маршрутизации, сравнивая адреса получателя пакета IPv6 с таблицей маршрутизации IPv6 маршрутизатора; соответствующий маршрут указывает направление дальнейшей передачи пакета IPv6.

ВНИМАНИЕ!

Если в приведенном выше списке заменить каждое слово IPv6 на IPv4, то все утверждения по-прежнему будут справедливы.

Хотя некоторые из приведенных выше концепций должны быть знакомы по протоколу IPv4, ниже приведено несколько рисунков, демонстрирующих их на примерах. Сначала на рис. 25.4 показано несколько параметров на хосте. У хоста (PC1) есть адрес 2345::1. Ему известен также его стандартный шлюз 2345::2. (Оба значения — допустимые сокращения для реальных IPv6-адресов.) Чтобы послать пакет IPv6 хосту PC2, расположенному в другой подсети IPv6, хост PC1 передает пакет IPv6 на маршрутизатор R1, являющийся стандартным шлюзом для хоста PC1.

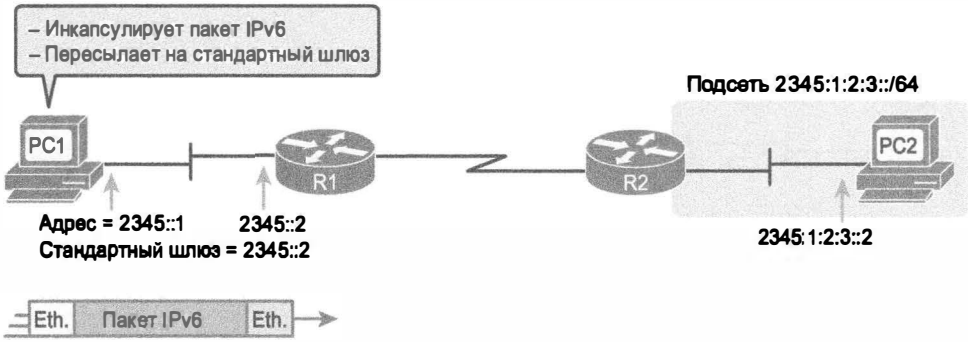


Рис. 25.4. Хост IPv6 создает и посылает пакет IPv6

При перенаправлении пакета IPv6 у маршрутизатора (R1) довольно много небольших задач, но пока сосредоточимся на работе, выполняемой маршрутизатором R1 в связи с инкапсуляцией. Как свидетельствует рис. 25.5, на первом этапе маршрутизатор R1 получает входящий фрейм канала связи и извлекает (деинкапсулирует) пакет IPv6 из фрейма, отбрасывает первоначальный заголовок и концевик канала связи. На втором этапе, как только маршрутизатор R1 узнает, что перенаправить пакет IPv6 следует на маршрутизатор R2, он добавляет к пакету IPv6 соответствующий заголовок и концевик исходящего канала связи, инкапсулируя пакет IPv6.

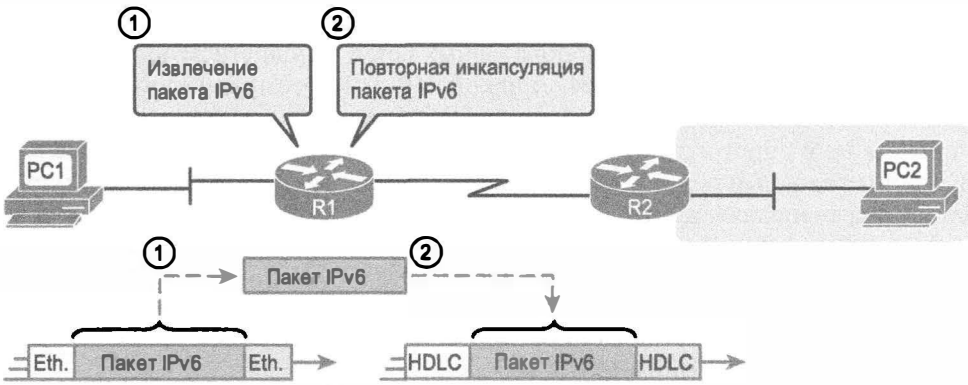


Рис. 25.5. При передаче пакета IPv6 маршрутизатор осуществляет стандартные задачи по инкапсуляции

Когда такой маршрутизатор, как R1, извлекает пакет из фрейма канала связи, он должен также выяснить, какой пакет находится во фрейме. Для этого маршрутиза-

тор должен просмотреть поле типа протокола в заголовке канала связи, идентифицирующее тип пакета во фрейме канала связи. В настоящее время большинство фреймов канала связи содержат либо пакет IPv4, либо IPv6.

Чтобы перенаправить пакет IPv6, маршрутизатор должен использовать свою таблицу маршрутизации IPv6, а не таблицу маршрутизации IPv4. Маршрутизатор должен просмотреть IPv6-адрес получателя пакета и сравнить его с текущей таблицей маршрутизации IPv6. Маршрутизатор использует инструкции перенаправления соответствующего маршрута IPv6 для перенаправления пакета IPv6. Общий процесс представлен на рис. 25.6.

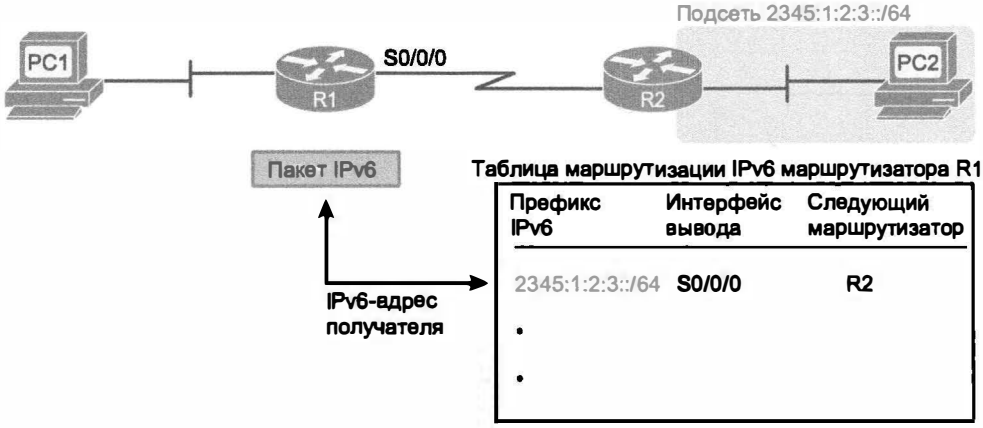


Рис. 25.6. При передаче пакета IPv6 маршрутизатор осуществляет стандартные задачи по инкапсуляции

Обратите внимание: процесс снова аналогичен таковому у протокола IPv4, за исключением того, что пакет IPv6 представляет IPv6-адреса для поиска в таблице маршрутизации IPv6, содержащей информацию о маршрутизации для подсетей IPv6 (называемую префиксами).

И наконец, в большинстве корпоративных сетей маршрутизаторы перенаправляют одновременно и пакеты IPv4, и пакеты IPv6. Поэтому компаниям не нужно принимать решение о переходе на протокол IPv6, скажем, в полночь воскресенья и полном отказе от протокола IPv4 на всех устройствах. Вместо этого протокол IPv6 допускает переходный период, на протяжении которого некоторые или все маршрутизаторы перенаправляют и пакеты IPv6, и IPv4. (Такая стратегия перехода называется *двойным стеком* (dual stack).) Все, что для этого необходимо, — это настроить маршрутизатор на перенаправление пакетов IPv6 в дополнение к существующей конфигурации для перенаправления пакетов IPv4.

Протоколы маршрутизации IPv6

Маршрутизаторы IPv6 должны изучить маршруты для всех возможных префиксов IPv6 (подсетей). Как и маршрутизаторы IPv4, маршрутизаторы IPv6 используют протоколы маршрутизации со знакомыми именами и, по правде говоря, со знакомыми функциями.

Ни один из протоколов маршрутизации IPv4 первоначально не предназначался для анонсирования маршрутов IPv6. Все они потребовали обновления для поддержки сообщений, протоколов и правил протокола IPv6. На протяжении хоть и довольно продолжительного времени протокол маршрутной информации (RIP), открытый протокол поиска первого кратчайшего маршрута (OSPF), расширенный протокол маршрутизации внутреннего шлюза (EIGRP) и протокол граничного шлюза (BGP) были модифицированы для поддержки протокола IPv6. Список имен этих протоколов маршрутизации с примечаниями приведен в табл. 25.1.

Таблица 25.1. Протоколы маршрутизации IPv6

Протокол маршрутизации	Определен	Примечания
RIPng (RIP Next Generation)	RFC	"Next Generation" (следующее поколение) — это ссылка на телевизионный сериал "Star Trek: the Next Generation" (Звездный путь)
OSPFv3 (OSPF версии 3)	RFC	Протокол OSPF, с которым работал протокол IPv4, фактически является версией 2, поэтому для протокола IPv6 была создана новая версия — OSPFv3
EIGRPv6 (EIGRP для IPv6)	Cisco	Права на протокол EIGRP принадлежат компании Cisco, но теперь она опубликовала его как информационный документ RFC
MP BGP-4 (Multiprotocol BGP версии 4)	RFC	Протокол BGP версии 4 был создан весьма растяжимым; для поддержки протокола IPv6 достаточно было добавить к версии 4 протокола BGP одно расширение и получить протокол MP BGP-4

Кроме того, эти протоколы маршрутизации следуют тем же соглашениям IGP и EGP, что и их версии для протокола IPv4. Протоколы RIPng, EIGRPv6 и OSPFv3 действуют как протоколы маршрутизации внутреннего шлюза, анонсируя маршруты IPv6 на предприятии.

Как уже упоминалось, протокол IPv6 использует большинство тех же концепций, что и протокол IPv4. Оба определяют заголовки с адресом отправителя и получателя, а также маршрутизацию пакетов, в процессе которой отбрасываются старые заголовки и концевики канала связи. Маршрутизаторы используют тот же общий процесс принятия решения о маршрутизации за счет сравнения IP-адреса получателя пакета с таблицей маршрутизации.

Наибольшее различие между протоколами IPv4 и IPv6 заключается в величине IPv6-адресов. В следующем разделе начинается рассмотрение специфических особенностей IPv6-адресов.

Адресация IPv6, формат и соглашения

Экзамены CCENT и CCNA R/S требуют наличия фундаментальных навыков в работе с IPv4-адресами. Например, необходимо уметь интерпретировать такие IPv4-адреса, как 172.21.73.14, работать с такими масками в префиксном стиле, как /25, и объяснить, что это означает при использовании с конкретным IPv4-адресом.

Необходимо также уметь найти идентификатор подсети по такому адресу и маске, как 172.21.73.14/25.

В данном разделе обсуждаются те же концепции, но для IPv6-адресов. В частности, рассматривается следующее.

- Как записать и интерпретировать несокращенные IPv6-адреса с 32 цифрами.
- Как сокращать и интерпретировать сокращенные IPv6-адреса.
- Как интерпретировать маску длины префикса IPv6.
- Как найти префикс IPv6 (идентификатор подсети) на основании адреса и маски длины префикса.

Самой большой проблемой здесь является размер чисел. К счастью, математический механизм вычисления идентификатора (зачастую довольно сложный для IPv4) намного проще для протокола IPv6, по крайней мере с точки зрения тем этой книги.

Представление полных (несокращенных) IPv6-адресов

Протокол IPv6 использует удобный шестнадцатеричный (hex) формат адресов. Чтобы сделать его более читаемым, протокол IPv6 использует формат с восемью наборами из четырех шестнадцатеричных цифр, разделенных двоеточием. Например: `2340:1111:AAAA:0001:1234:5678:9ABC:1234`

ВНИМАНИЕ!

Для удобства описания набора из четырех шестнадцатеричных цифр автор использует термин *квартет* (quartet). Каждый IPv6-адрес состоит из восьми квартетов. Обратите внимание, что запросы на комментарии IPv6 не используют термин *квартет*.

У IPv6-адресов есть также двоичный формат, но, к счастью, двоичную версию адресов приходится использовать не часто. Но даже и в этих случаях преобразование из шестнадцатеричного формата в двоичный осуществляется относительно просто. Достаточно заменить каждую шестнадцатеричную цифру на эквивалентное 4-битовое значение, представленное в табл. 25.2.

Таблица 25.2. Шестнадцатерично-двоичные преобразования

Шестнадцатеричное	Двоичное	Шестнадцатеричное	Двоичное
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Сокращение и расширение IPv6-адресов

Протокол IPv6 определяет также способы сокращения IPv6-адресов при написании и вводе. Почему? Хотя использование шестнадцатеричных чисел с 32 цифрами намного удобней, чем 128-битовых двоичных чисел, 32 шестнадцатеричных цифры — это все еще слишком много для запоминания, распознавания в командах вывода и ввода в командной строке. Правила сокращения IPv6-адреса позволяют сокращать эти числа.

Компьютеры и маршрутизаторы обычно используют самое краткое сокращение, даже если ввести все 32 цифры шестнадцатеричного адреса. Так, даже если вы предпочитаете использовать более длинную, несокращенную версию IPv6-адреса, необходимо быть готовым интерпретировать значение сокращенного IPv6-адреса, используемого маршрутизаторами и хостами. В данном разделе сначала рассматривается сокращение, а затем расширение адресов.

Сокращение IPv6-адресов

Ниже приведены два простых правила, позволяющих любому человеку или компьютеру сокращать IPv6-адреса.

Правила сокращения IPv6-адресов

Ключевая тема

1. В каждом квартете из четырех шестнадцатеричных цифр удалите предваряющие нули в трех позициях слева (нули с левой стороны квартета). (Примечание: на этом этапе от квартета 0000 останется один 0.)
2. Найдите все строки по два или более последовательных квартета из всех нулей и замените этот набор квартетов двойным двоеточием (`::`). Оно означает "два или более квартета из всех нулей". Однако двойное двоеточие можно использовать в адресе только однажды, поскольку в противном случае IPv6-адрес мог быть неоднозначен.

Рассмотрим, например, следующий IPv6-адрес.

```
FE00:0000:0000:0001:0000:0000:0000:0056
```

Первое правило применимо ко всем восьми квартетам независимо. В каждом удалите все предваряющие нули. Обратите внимание, что у пяти квартетов есть по четыре нуля, поэтому удалите для них только по три нуля, оставив один. Получится следующее значение:

```
FE00:0:0:1:0:0:0:56
```

Несмотря на то что это сокращение вполне допустимо, адрес может быть сокращен еще больше при помощи второго правила. В данном случае есть два фрагмента, где у нескольких квартетов подряд есть только нули. Выберите самую длинную последовательность и замените ее на `::`, получив самое короткое допустимое сокращение:

```
FE00:0:0:1::56
```

Хотя сокращение `FE00:0:0:1::56` действительно является самым коротким, этот пример позволяет увидеть две наиболее распространенные ошибки сокращения IPv6-адресов. Во-первых, никогда не удаляйте замыкающие нули в квартете (0 с правой стороны квартета). В данном случае первый квартет, `FE00`, не может быть сокра-

щен вообще, поскольку два нуля находятся в конце. Поэтому следующий адрес, только со значением FE в первом квартете, не является правильным сокращением первоначального IPv6-адреса:

```
FE:0:0:1::56
```

Вторая распространенная ошибка — замена всех последовательностей нулевых квартетов двойными двоеточиями. Например, следующее сокращение было бы неправильным для первоначального IPv6-адреса, приведенного выше:

```
FE00::1::56
```

Это сокращение неправильно потому, что теперь невозможно установить точное количество квартетов с нулями, замененных каждым знаком :: в первоначальном несокращенном адресе.

Расширение сокращенных IPv6-адресов

Чтобы развернуть IPv6-адрес в его полное несокращенное значение из 32 цифр, используйте два подобных правила. Эти правила основаны на обратной логике предыдущих двух правил.

Ключевая тема **Правила развертывания сокращенных IPv6-адресов**

1. В каждом квартете добавьте предваряющие нули, если необходимо, пока у квартета не будет четырех шестнадцатеричных цифр.
2. Если есть двойное двоеточие (::), подсчитывайте представленные в настоящее время квартеты. Поскольку общее количество квартетов должно быть равно 8, замените :: недостающим количеством квартетов 0000.

Лучше всего освоиться с этими адресами и сокращениями на практике. В табл. 25.3 приведен список из нескольких практических заданий с полным IPv6-адресом на 32 цифры слева и наилучшим сокращением справа. Представлен либо развернутый, либо сокращенный адрес, а записать необходимо противоположное значение. Ответы приведены в конце главы, в разделе "Ответы на практические задания главы".

Таблица 25.3. Практика сокращения и развертывания IPv6-адреса

Полный	Сокращенный
2340:0000:0010:0100:1000:ABCD:0101:1010	
	30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009
2222:3333:4444:5555:0000:0000:6060:0707	
	3210::
210F:0000:0000:0000:CCCC:0000:0000:000D	
	34BA:B:B::20
FE80:0000:0000:0000:DEAD:BEFF:FEEF:CAFE	
	FE80::FACE:BAFF:FEBE:CAFE
FE80:000F:00E0:0D00:FACE:BAFF:FE00:0000	
	FE80:800:0:40:CAFE:FF:FE00:1

По мере прохождения практики появятся навыки работы с сокращениями. Раздел "Практика" в конце этой главы содержит несколько рекомендаций по дальнейшему приобретению практических навыков.

Представление длины префикса адреса

Протокол IPv6 использует концепцию маски под названием *длина префикса* (prefix length), подобную маскам подсети IPv4. Как и маска префиксного стиля протокола IPv4, длина префикса протокола IPv6 записывается как символ /, сопровождаемый десятичным числом. Длина префикса определяет количество битов префикса IPv6-адреса. Эта концепция в основном совпадает с концепцией идентификатора подсети IPv4.

При записи IPv6-адресов, если длина префикса имеет значение, она указывается за IPv6-адресом. В документации можно оставить пробел между адресом и знаком /, но при вводе значений во время настройки маршрутизатора Cisco пробел необязателен. Для адреса с 64-битовой длиной префикса применима любая из этих форм:

```
2222:1111:0:1:A:B:C:D/64
2222:1111:0:1:A:B:C:D /64
```

И наконец, обратите внимание на то, что длина префикса составляет несколько битов, поэтому допустимым будет диапазон значений IPv6-адресов от 0 до 128 включительно.

Вычисление префикса IPv6 (идентификатора подсети)

В протоколе IPv4 можно взять IP-адрес и связанную с ним маску подсети и вычислить идентификатор подсети. В подсетях IPv6 можно взять IPv6-адрес и связанную с ним длину префикса и вычислить эквивалент идентификатора подсети IPv6: *префикс IPv6* (IPv6 prefix).

Подобно маскам подсети IPv4, некоторые длины префикса IPv6 упрощают математические вычисления, а другие затрудняют. В этом разделе рассматриваются более простые случаи главным образом потому, что размер IPv6-адреса позволяет всем выбирать длины префикса IPv6 так, чтобы упростить математические вычисления.

Вычисление префикса IPv6

В протоколе IPv6 префикс представляет группу IPv6-адресов. Пока этот раздел сосредоточивается на математике и только на математике вычисления чисел, представляющих этот префикс. Фактические значения рассматриваются в главе 26.

Значение каждого префикса IPv6 (или подсети, если так больше нравится) представляет группу. В запросах на комментарии IPv6 само число также называется префиксом, но многие продолжают называть его номером или идентификатором подсети, используя те же термины, что и в протоколе IPv4. Подобно протоколу IPv4, можно начать с IPv6-адреса и длины префикса, а затем найти префикс, используя те же общие правила, что и протокол IPv4. Если префикс имеет длину /P, то используйте следующие правила.

Ключевая тема

Этапы процесса вычисления префикса IPv6 на основании IPv6-адреса и длины префикса

1. Скопируйте первые биты P .
2. Остальную часть битов замените нулями.

При использовании длины префикса, которая случайно оказывается кратной 4, можно думать не в терминах битов, а в терминах шестнадцатеричных цифр. Кратная 4, длина префикса означает, что каждая шестнадцатеричная цифра или копируется, или заменяется нулями. Только для справки: если длина префикса действительно кратна 4, процесс становится таким.

1. Определите количество шестнадцатеричных цифр в префиксе, разделив длину префикса (который представлен в битах) на 4.
2. Скопируйте шестнадцатеричные цифры, определенные в префиксе на первом этапе.
3. Остальная часть шестнадцатеричных цифр заменяется нулями.

На рис. 25.7 приведен пример с длиной префикса 64. В данном случае на этапе 1 рассматривается длина префикса /64 и вычисляется наличие у префикса 16-ти шестнадцатеричных цифр. На этапе 2 копируются первые 16 цифр IPv6-адреса, а на этапе 3 записываются шестнадцатеричные нули для остальной части цифр.

Рис. 25.7. Получение префикса IPv6 из адреса и длины

После нахождения префикса IPv6 следует быть готовым к сокращению префикса IPv6 по тем же правилам, что и для сокращения IPv6-адреса. Но дополнительное внимание следует обратить на конец префикса, поскольку там зачастую есть несколько октетов, заполненных всеми нулями. В результате сокращение обычно завершается двумя двоеточиями (::).

Рассмотрим, например, следующий IPv6-адрес, присвоенный хосту в сети LAN:

```
2000:1234:5678:9ABC:1234:5678:9ABC:1111/64
```

В этом примере представлен IPv6-адрес, который сам по себе не может быть сокращен. После вычисления префикса для подсети, в которой располагается адрес, после обнуления последних 64 битов (16 цифр) адреса получается следующее префиксное значение:

```
2000:1234:5678:9ABC:0000:0000:0000:0000/64
```

Это значение может быть сокращено на четыре нулевых квартета в конце следующим образом:

```
2000:1234:5678:9ABC::/64
```

Для приобретения навыков в вычислениях уделите время выполнению ряда практических заданий, представленных в табл. 25.4. Ответы приведены в конце главы в разделе "Ответы на практические задания главы".

Таблица 25.4. Вычисление префикса IPv6 по значениям адреса и длины

Адрес/длина	Префикс
2340:0:10:100:1000:ABCD:101:1010/64	
30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009/64	
2222:3333:4444:5555::6060:707/64	
3210::ABCD:101:1010/64	
210F::CCCC:B0B0:9999:9009/64	
34BA:B:B:0:5555:0:6060:707/64	
3124::DEAD:CAFE:FF:FE00:1/64	
2BCD::FACE:BEFF:FEBE:CAFE/64	
3FED:F:E0:D00:FACE:BAFF:FE00:0/64	
3BED:800:0:40:FACE:BAFF:FE00:0/64	

Раздел "Практика" в конце этой главы содержит несколько рекомендаций по дальнейшему приобретению практических навыков. Раздел "Ответы на практические задания главы" в конце главы содержит также табл. 25.8, в которой приведена законченная версия этой таблицы, чтобы можно было проверить свою работу.

Работа с более трудными длинами префикса IPv6

Некоторые длины префикса упрощают вычисление префикса, а некоторые требуют применения двоичной математики. Если длина префикса кратна 16, процесс копирования части адреса задействует квартеты целиком. Если длина префикса кратна не 16, а 4, то граница, по крайней мере, находится на краю шестнадцатеричной цифры, и можно избежать применения двоичных вычислений.

Хотя длина префикса /64, безусловно, наиболее распространенная, следует быть готовым к вычислению префикса, длина которого не кратна 4. Рассмотрим, например, следующий IPv6-адрес и длину префикса:

```
2000:1234:5678:9ABC:1234:5678:9ABC:1111/56
```

Поскольку в этом примере используется длина префикса /56, префикс включает первые 56 битов или 14 первых полных шестнадцатеричных цифр адреса. Остальная часть шестнадцатеричных цифр будет нулями, что даст следующий префикс:

```
2000:1234:5678:9A00:0000:0000:0000:0000/56
```

Это значение может быть сокращено на четыре нулевых квартета в конце следующим образом:

```
2000:1234:5678:9A00::/56
```

Данный пример демонстрирует случай весьма распространенной ошибки. Иногда люди видят длину /56 и представляют себе первые 14 шестнадцатеричных цифр, что совершенно верно. Затем они копируют первые 14 шестнадцатеричных цифр и добавляют парное двоеточие, получая следующее:

```
2000:1234:5678:9A::/56
```

Это сокращение неправильное, так как оно потеряло замыкающие нули (00) в конце четвертого квартета. Когда граница проходит не по краю квартета, будьте внимательны при сокращении.

Здесь снова поможет дополнительная практика. Для дополнительной практики примеры в табл. 25.5 предлагают длину префикса, кратную 4, но не проходящую по границе квартета. Ответы приведены в конце главы, в разделе "Ответы на практические задания главы".

Таблица 25.5. Вычисление префикса IPv6 по значениям адреса и длины

Адрес/длина	Префикс
34BA:B:B:0:5555:0:6060:707/80	
3124::DEAD:CAFE:FF:FE00:1/80	
2BCD::FACE:BEFF:FEBE:CAFE/48	
3FED:F:E0:D00:FACE:BAFF:FE00:0/48	
210F:A:B:C:CCCC:B0B0:9999:9009/40	
34BA:B:B:0:5555:0:6060:707/36	
3124::DEAD:CAFE:FF:FE00:1/60	
2BCD::FACE:1:BEFF:FEBE:CAFE/56	
3FED:F:E0:D000:FACE:BAFF:FE00:0/52	
3BED:800:0:40:FACE:BAFF:FE00:0/44	

Обзор

Резюме

- Главная цель базового протокола IPv6 та же, что и у протокола IPv4. Базовый протокол IPv6 определен в документе RFC 2460 как концепция пакета, адреса этих пакетов, а также роли хостов и маршрутизаторов. Установленные правила позволяют устройствам перенаправлять пакеты хостам через несколько маршрутизаторов так, чтобы они достигли правильного хоста получателя.
- Поскольку протокол IPv6 влияет на очень многие функции сети TCP/IP, было создано еще несколько запросов на комментарии, определяющих связанные с ним подробности. Еще несколько запросов на комментарии определяют, как осуществляется переход с протокола IPv4 на IPv6, а также новые версии знакомых протоколов или новые протоколы на замену старым. Соответствующие примеры приведены ниже.
  - **Прежняя версия 2 протокола OSPF обновлена до версии 3**. Протокол OSPF версии 2 годился для протокола IPv4, но не для IPv6, поэтому для его поддержки была создана более новая версия, OSPF версии 3.
  - **Протокол ICMP обновлен до версии 6**. Протокол управляющих сообщений Интернета (ICMP) годился для протокола IPv4, но для поддержки протокола IPv6 его пришлось изменить. Новое имя протокола — ICMPv6.
  - **Протокол ARP заменен протоколом обнаружения соседних устройств**. Для протокола IPv4 используемые соседями MAC-адреса обнаруживает протокол преобразования адресов (ARP). Протокол IPv6 заменяет протокол ARP более общим протоколом обнаружения соседних устройств (NDP).
- Хотя термин IPv6 используется довольно широко и включает множество протоколов, новый 128-битовый IPv6-адрес определяет один конкретный протокол под названием IPv6.
- Подобно многим другим функциям, маршрутизация IPv6 выглядит точно так же, как и маршрутизация IPv4, но с различиями в специфических подробностях. Сначала обсудим концепции, общие для протоколов IPv6 и IPv4.
  - Чтобы интерфейс устройства конечного пользователя мог создавать и передавать пакеты IPv6, он нуждается в IPv6-адресе.
  - Хостам конечного пользователя должен быть известен IPv6-адрес стандартного маршрутизатора, на который хост посылает пакеты IPv6, если хост получателя находится в другой подсети.
  - При передаче пакета маршрутизаторы IPv6 извлекают и повторно инкапсулируют каждый пакет IPv6.
  - Маршрутизаторы IPv6 принимают решение о маршрутизации сравнивая адреса получателя пакета IPv6 с таблицей маршрутизации IPv6 маршрутизатора; соответствующий маршрут указывает направление дальнейшей передачи пакета IPv6.

- Протокол IPv6 использует удобный шестнадцатеричный (hex) формат адресов. Чтобы сделать его более читаемым, протокол IPv6 использует формат с восемью наборами из четырех шестнадцатеричных цифр, разделенных двоеточием. Например:

  `2340:1111:AAAA:0001:1234:5678:9ABC:1234`

- Ниже приведены два простых правила, позволяющих любому человеку или компьютеру сокращать IPv6-адреса.
  - В каждом квартете из четырех шестнадцатеричных цифр удалите предваряющие нули в трех позициях слева (нули с левой стороны квартета). (Примечание: на этом этапе от квартета 0000 останется один 0.)
  - Найдите все строки по два или более последовательных квартета из всех нулей и замените этот набор квартетов двойным двоеточием (`::`). Оно означает "два или более квартета из всех нулей". Однако двойное двоеточие можно использовать в адресе только однажды, поскольку в противном случае IPv6-адрес мог быть неоднозначен.
- Чтобы развернуть IPv6-адрес в его полное несокращенное значение из 32 цифр, используйте два подобных правила. Эти правила основаны на обратной логике предыдущих двух правил.
  - В каждом квартете добавьте предваряющие нули, если необходимо, пока у квартета не будет четырех шестнадцатеричных цифр.
  - Если есть двойное двоеточие (`::`), подсчитывайте представленные в настоящее время квартеты. Поскольку общее количество квартетов должно быть равно 8, замените :: недостающим количеством квартетов 0000.
- Протокол IPv6 использует концепцию маски под названием *длина префикса*, подобную маскам подсети IPv4. Как и маска префиксного стиля протокола IPv4, длина префикса протокола IPv6 записывается как символ /, сопровождаемый десятичным числом. Длина префикса определяет количество битов префикса IPv6-адреса. Это в основном та же концепция, что и идентификатор подсети IPv4.
- Подобно протоколу IPv4, можно начать с IPv6-адреса и длины префикса, а затем найти префикс, используя те же общие правила, что и протокол IPv4. Если префикс имеет длину /P, то используйте следующие правила.
  - Скопируйте первые биты P.
  - Остальную часть битов замените нулями.
- При использовании длины префикса, которая случайно оказывается кратной 4, можно думать не в терминах битов, а в терминах шестнадцатеричных цифр. Кратная 4, длина префикса означает, что каждая шестнадцатеричная цифра или копируется, или заменяется нулями. Только для справки: если длина префикса действительно кратна 4, процесс становится таким.
  - Определите количество шестнадцатеричных цифр в префиксе, разделив длину префикса (который представлен в битах) на 4.

- Скопируйте шестнадцатеричные цифры, определенные в префиксе на первом этапе.
- Остальная часть шестнадцатеричных цифр заменяется нулями.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении В на веб-сайте.

1. Что из следующего было краткосрочным решением проблемы исчерпания IPv4-адресов?

   А) IP версии 6.

   Б) IP версии 5.

   В) NAT/PAT

   Г) ARP.

2. Маршрутизатор получает фрейм Ethernet, содержащий пакет IPv6. Затем он принимает решение о перенаправлении пакета последовательного канала связи. Какое из следующих утверждений описывает перенаправление маршрутизатором пакета IPv6?

   А) Маршрутизатор отбрасывает заголовок и концевик канала связи Ethernet полученного фрейма.

   Б) Маршрутизатор принимает решение о перенаправлении на основании исходящего IPv6-адреса пакета.

   В) Маршрутизатор сохраняет заголовок Ethernet, инкапсулируя весь фрейм в новом пакете IPv6, прежде чем послать его по последовательному каналу связи.

   Г) При выборе направления перенаправления пакета маршрутизатор использует таблицу маршрутизации IPv4.

3. Что из приведенного ниже является правильным сокращением записи адреса FE80:0000:0000:0100:0000:0000:0000:0123?

   А) FE80::100::123.

   Б) FE8::1::123.

   В) FE80::100:0:0:0:123:4567.

   Г) FE80:0:0:100::123.

4. Что из следующего является самым коротким допустимым сокращением адреса 2000:0300:0040:0005:6000:0700:0080:0009?

   А) 2:3:4:5:6:7:8:9.

   Б) 2000:300:40:5:6000:700:80:9.

   В) 2000:300:4:5:6000:700:8:9.

   Г) 2000:3:4:5:6:7:8:9.

5. Что из следующего является несокращенной версией IPv6-адреса 2001:DB8::200:28?

   А) 2001:0DB8:0000:0000:0000:0000:0200:0028.

   Б) 2001:0DB8::0200:0028.

В) 2001:0DB8:0:0:0:0:0200:0028.

Г) 2001:0DB8:0000:0000:0000:0000:200:0028.

6. Что из следующего является префиксом для адреса 2000:0000:0000:0005:6000:0700:0080:0009 при маске /64?

А) 2000::5::/64.

Б) 2000::5:0:0:0:0/64.

В) 2000:0:0:5::/64.

Г) 2000:0:0:5:0:0:0:0/64.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 25.6.

Таблица 25.6. Ключевые темы главы 25

Элемент	Описание	Страница
Список	Подобия протоколов IPv4 и IPv6	738
Список	Правила сокращения IPv6-адресов	743
Список	Правила развертывания сокращенных IPv6-адресов	744
Список	Этапы процесса вычисления префикса IPv6 на основании IPv6-адреса и длины префикса	746

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

исчерпание IPv4-адресов (IPv4 address exhaustion), IETF, NAT, CIDR, протокол IP версии 6 (IP version 6 — IPv6), протокол OSPF версии 3 (OSPF version 3 — OSPFv3), протокол EIGRP версии 6 (EIGRP version 6 — EIGRPv6), префикс (prefix), длина префикса (prefix length), квартет (quartet)

Практика

Ответы на практические задания главы

В этом разделе содержатся ответы на практические задания, приведенные в разных разделах главы. Ответы см. в табл. 25.7–25.9.

Таблица 25.7. Ответы на вопросы, приведенные в табл. 25.3

Полный	Сокращенный
2340:0000:0010:0100:1000:ABCD:0101:1010	2340:0:10:100:1000:ABCD:101:1010
30A0:ABCD:EF12:3456:0ABC:B0B0:9999:9009	30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009
2222:3333:4444:5555:0000:0000:6060:0707	2222:3333:4444:5555::6060:707
3210:0000:0000:0000:0000:0000:0000:0000	3210::
210F:0000:0000:0000:CCCC:0000:0000:000D	210F::CCCC:0:0:D
34BA:000B:000B:0000:0000:0000:0000:0020	34BA:B:B::20
FE80:0000:0000:0000:DEAD:BEFF:FEEF:CAFE	FE80::DEAD:BEFF:FEEF:CAFÉ
FE80:0000:0000:0000:FACE:BAFF:FEBE:CAFE	FE80::FACE:BAFF:FEBE:CAFÉ
FE80:000F:00E0:0D00:FACE:BAFF:FE00:0000	FE80:F:E0:D00:FACE:BAFF:FE00:0
FE80:0800:0000:0040:CAFE:00FF:FE00:0001	FE80:800:0:40:CAFE:FF:FE00:1

Таблица 25.8. Ответы на вопросы, приведенные в табл. 25.4

Адрес/длина	Префикс
2340:0:10:100:1000:ABCD:101:1010/64	2340:0:10:100::/64
30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009/64	30A0:ABCD:EF12:3456::/64
2222:3333:4444:5555::6060:707/64	2222:3333:4444:5555::/64
3210::ABCD:101:1010/64	3210::/64
210F::CCCC:B0B0:9999:9009/64	210F::/64
34BA:B:B:0:5555:0:6060:707/64	34BA:B:B::/64
3124::DEAD:CAFE:FF:FE00:1/64	3124:0:0:DEAD::/64
2BCD::FACE:BEFF:FEBE:CAFE/64	2BCD::/64
3FED:F:E0:D00:FACE:BAFF:FE00:0/64	3FED:F:E0:D00::/64
3BED:800:0:40:FACE:BAFF:FE00:0/64	3BED:800:0:40::/64

Таблица 25.9. Ответы на вопросы, приведенные в табл. 25.5

Адрес/длина	Префикс
34BA:B:B:0:5555:0:6060:707/80	34BA:B:B:0:5555::/80
3124::DEAD:CAFE:FF:FE00:1/80	3124:0:0:DEAD:CAFE::/80
2BCD::FACE:BEFF:FEBE:CAFE/48	2BCD::/48
3FED:F:E0:D00:FACE:BAFF:FE00:0/48	3FED:F:E0::/48
210F:A:B:C:CCCC:B0B0:9999:9009/40	210F:A::/40
34BA:B:B:0:5555:0:6060:707/36	34BA:B::/36
3124::DEAD:CAFE:FF:FE00:1/60	3124:0:0:DEA0::/60
2BCD::FACE:1:BEFF:FEBE:CAFE/56	2BCD:0:0:FA00::/56
3FED:F:E0:D000:FACE:BAFF:FE00:0/52	3FED:F:E0:D000::/52
3BED:800:0:40:FACE:BAFF:FE00:0/44	3BED:800::/44

Ответы на контрольные вопросы:
1 В. **2** А. **3** Г. **4** Б. **5** А. **6** В.

ГЛАВА 26

IPv6-адресация и создание подсетей

Протокол IPv4 организует пространство адресов несколькими способами. В первую очередь он разделяет адреса по классам, на одноадресатные IPv4-адреса классов A, B и C. (Термин *одноадресатный* (unicast) означает тот факт, что каждый адрес используется только одним интерфейсом.) Далее, в рамках адресных интервалов класса A, B и C, Агентство по назначению адресов Интернета (Internet Assigned Numbers Authority — IANA) и Ассоциация по присвоению имен и номеров портов Интернета (Internet Corporation for Assigned Names and Numbers — ICANN) резервируют большую часть адресов как открытые IPv4-адреса, а некоторые резервируют как частные.

Протокол IPv6 не использует концепцию классовой сети, как протокол IPv4. Хотя агентство IANA все еще резервирует некоторый диапазон IPv6-адресов в специфических целях и даже некоторые интервалы адресов, служащие открытыми и частными IPv6-адресами. Кроме того, агентство IANA попыталось проявить практический подход к резервированию диапазонов всех пространств IPv6-адресов для разных целей, — они учли опыт, полученный за нескольких десятилетий быстрого развития Интернета.

В этой главе два главных раздела. В первом рассматриваются *глобальные одноадресатные адреса* (global unicast address), являющиеся открытыми IPv6-адресами. Второй раздел посвящен *уникальным локальным адресам* (unique local address), служашим частными IPv6-адресами.

В этой главе рассматриваются следующие экзаменационные темы

IP-адресация (IPv4/IPv6)

Выбор подходящей схемы IPv6-адресации, удовлетворяющей требованиям адресации в среде LAN/WAN.

Описание IPv6-адреса.

Глобальный одноадресатный.

Уникальный локальный адрес.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Концепции глобальной одноадресатной адресации

В данном разделе основное внимание уделяется одному типу одноадресатных IPv6-адресов: глобальным одноадресатным адресам. Кроме того, большинство общих концепций и процессов, связанных с глобальными одноадресатными IPv6-адресами, аналогичны таковым у первоначальных открытых IPv4-адресов. Поэтому этот раздел начинается с обзора некоторых концепций протокола IPv4, сопровождаемого подробностями использования компаниями глобальных одноадресатных адресов.

В этом разделе обсуждаются также подсети IPv6 и весь процесс получения блока глобальных одноадресатных адресов и создания подсетей для одной компании. Процесс подразумевает получение глобально уникального префикса маршрутизации, создание подсети IPv6 и присвоение IPv6-адресов в каждой подсети, как и в протоколе IPv4.

Краткий обзор открытых и частных IPv4-адресов

Исторически IPv4-адресация начиналась с идеи, что каждому отдельному хосту будет присвоен глобально уникальный открытый IPv4-адрес. Но, как уже неоднократно упоминалось, IPv4-адресов оказалось недостаточно. Так, в 1990-х годах компании начали использовать адреса из диапазона частных IPv4-адресов, как определено в документе RFC 1918. Эти компании или просто не подключались к Интернету, или подключались с применением NAT, совместно используя несколько открытых глобально уникальных IPv4-адресов для всех подключений хостов к Интернету.

Ниже кратко рассматриваются некоторые из основных концепций использования открытых и частных IPv4-адресов по сравнению с эквивалентными адресами протокола IPv6.

Обзор концепций открытой IPv4-адресации

В первоначальном проекте Интернета предполагалось, что каждый хост IPv4 будет использовать уникальный одноадресатный адрес. Чтобы каждый одноадресатный адрес был уникален, в планировании должны быть выполнены три главных этапа.

- Компания или организация составляет запрос и получает право на исключительное использование открытых номеров IPv4 сети класса A, B или C.
- Инженеры компании разделяют эту классовую сеть на меньшие подсети, гарантируя использование каждой подсети только в одном месте компании.
- В каждой подсети инженеры назначают индивидуальные IPv4-адреса, гарантируя использование каждого адреса только для одного интерфейса хоста.

На рис. 26.1 приведено концептуальное представление разделения классовой сети IPv4 на подсети, содержащие индивидуальные одноадресатные IPv4-адреса. Здесь представлена вся открытая сеть класса A, B или C с небольшими прямоугольниками подсетей и индивидуальными одноадресатными IPv4-адресами в виде почтовых ящиков.

Кроме представленных на рис. 26.1 общих концепций разделения предприятием классовой сети IPv4 на подсети, следует также запланировать использование подсе-

тей в корпоративной объединенной сети. К настоящему времени эти идеи должны быть относительно знакомы, но для обзора технологий, рассматриваемых на сертификационных экзаменах CCENT и CCNA, ниже приведены элементы, испытывающие потребность в отдельных подсетях IPv4.

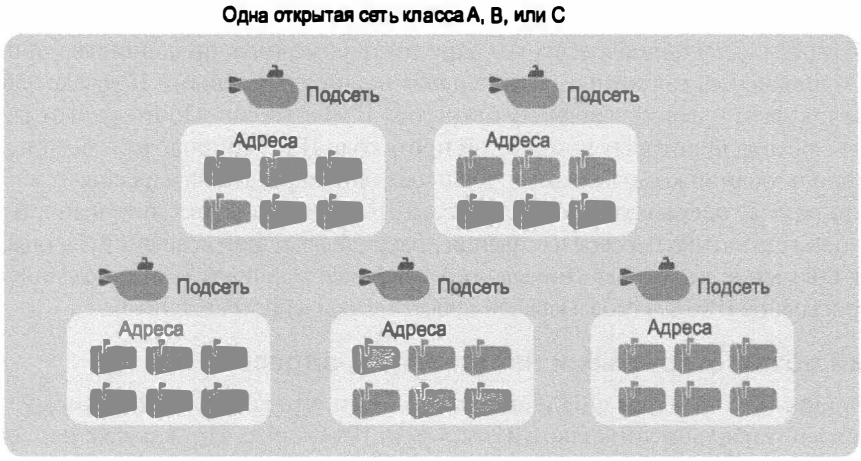


Рис. 26.1. Уникальная сеть IP, уникальные подсети и адреса в подсети

Ключевая тема

Сетевые каналы связи, нуждающиеся в подсетях IPv6

- Сети VLAN.
- Двухточечный последовательный канал связи.
- Эмуляция Ethernet канал связи WAN (EoMPLS).
- Frame Relay PVC (подробно обсуждается в книге по ICND2).

Например, в корпоративной объединенной сети, представленной на рис. 26.2, планируется создать пять подсетей. В этом примере каждый интерфейс LAN маршрутизатора подключен к сети LAN с использованием одной сети VLAN (в общей сложности для трех подсетей и трех сетей VLAN). Каждый последовательный канал связи и канал связи Ethernet WAN также испытывает потребность в подсети. (Для Интернета подсети назначаются различными провайдерами услуг Интернета.)

Обзор концепций частной IPv4-адресации

Откровенно говоря, большинство компаний не использует открытые IPv4-адресов в объединенных корпоративных сетях. Мир начал сталкиваться с проблемой исчерпания IPv4-адресов, и это потребовало некоторых изменений.

Сегодня большинство корпоративных объединенных сетей использует для большинства хостов частные IPv4-адреса. Дело в том, что использование частных IPv4-адресов (RFC 1918) наряду с NAT/PAT значительно сокращает количество необходимых организации открытых IPv4-адресов. Использование частных IPv4-адресов вместе с NAT/PAT позволяет одному открытому IPv4-адресу обеспечивать работу довольно большой корпоративной объединенной сети, отодвигая день, когда будут исчерпаны открытые IPv4-адреса. (Обзор некоторых из событий, обусловивших потребность в частных IPv4-адресах и NAT/PAT, см. в главе 25.)

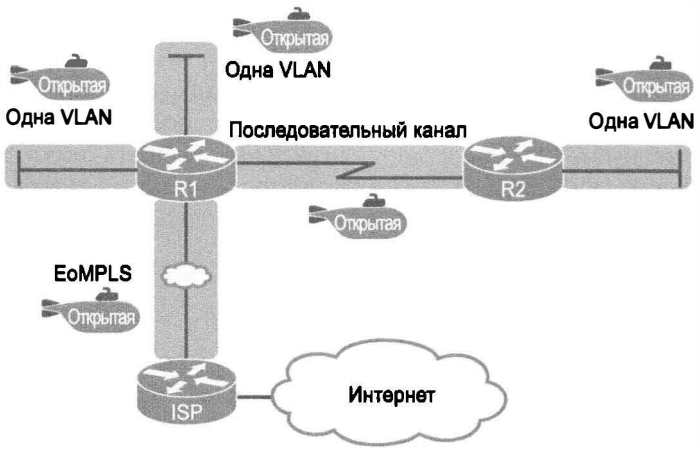


Рис. 26.2. Пример объединенной сети с пятью подсетями IPv4 с открытыми адресами

Для сравнения на рис. 26.3 показан тот же проект корпоративной объединенной сети, представленный на рис. 26.3. Но в данном случае предприятие использует частные IPv4-адреса в большей части сети, а маршрутизатор R1, поддерживающий NAT/PAT, сокращает количество необходимых открытых IPv4-адресов.

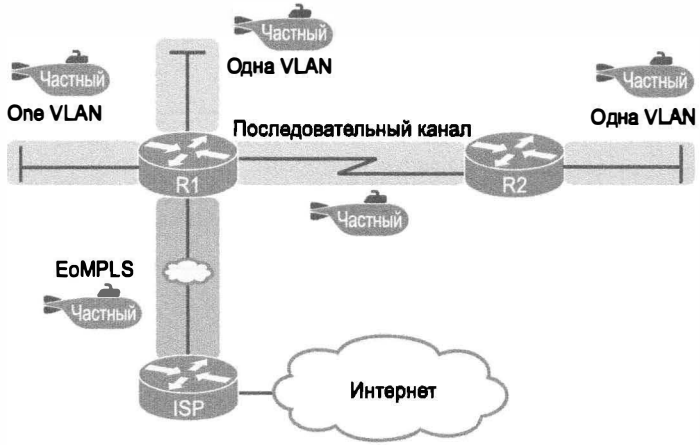


Рис. 26.3. Пример объединенной сети с пятью подсетями IPv4

Открытые и частные IPv6-адреса

Для одноадресатной адресации протокол IPv6 предоставляет две подобные возможности. Начнем с *глобальных одноадресатных адресов* (global unicast address), которые являются аналогом первоначальных открытых адресов протокола IPv4. Подобно открытым IPv4-адресам, глобальные одноадресатные IPv6-адреса полагаются на административный процесс присвоения каждой компании блока уникальных IPv6-адресов. Затем каждая компания разделяет этот блок IPv6-адресов на подсети и использует адреса только из этого блока. В результате компания использует адреса, уникальные и в мировом масштабе.

Вторая возможность — это *уникальные локальные адреса* (unique local address), подобные частным адресам протокола IPv4. Компании, не планирующие подключение к Интернету, и компании, планирующие использовать NAT IPv6, могут использовать эти частные уникальные локальные адреса. Процесс тот же, что и у IPv4: инженер может прочитать детали в документе RFC, выбрать подходящие номера и начать присвоение IPv6-адресов без необходимости регистрировать их в IANA или любой другой организации.

Ниже кратко резюмированы эти пункты.

Ключевая тема **Два типа одноадресатных адресов IPv6**

- **Глобальный одноадресатный адрес.** Аналог открытых IPv4-адресов. Организация, нуждающаяся в IPv6-адресах, запрашивает зарегистрированный блок IPv6-адресов, присваиваемый как глобальный префикс маршрутизации. После этого только данная организация использует адреса из этого блока, т.е. адреса, начинающиеся с присвоенного префикса.
- **Уникальный локальный адрес.** Аналог частных IPv4-адресов. Несколько организаций вполне могут использовать эти же адреса без необходимости регистрировать их в соответствующих инстанциях.

Далее в этом разделе более подробно рассматриваются глобальные одноадресатные адреса, а уникальные локальные адреса описаны в следующем разделе.

ВНИМАНИЕ!

Для полноты картины обратите внимание, что можно также найти документацию о другом диапазоне адресов, *локальном для площадки* (site local). Эти адреса определены префиксом FEC0::/10 (поэтому они начинаются на FEC, FED, FEE или FEF) и первоначально предназначались для использования в качестве частных адресов IPv4, но теперь они удалены из стандарта IPv6.

Глобальный префикс маршрутизации IPv6

Глобальные одноадресатные IPv6-адреса позволяют протоколу IPv6 работать подобно первоначальному проекту протокола IPv4. Другими словами, каждая организация запрашивает блок IPv6-адресов, которые никто больше не может использовать. Затем организация разделяет блок адресов на меньшие блоки, называемые подсетями. И наконец, для каждого конкретного хоста инженер выбирает подходящий IPv6-адрес из соответствующий подсети.

Этот зарезервированный блок IPv6-адресов может использовать только одна компания, поэтому он называется *глобальным префиксом маршрутизации* (global routing prefix). Каждая организация, которая собирается подключиться к Интернету и использовать глобальные одноадресатные IPv6-адреса, должна запросить и получить глобальный префикс маршрутизации. В самом общем случае глобальный префикс маршрутизации можно считать номером сети класса A, B или C из диапазона открытых IPv4-адресов.

Термин *глобальный префикс маршрутизации*, казалось бы, мало подходит для блока IPv6-адресов. Фактически термин относится к идее, что у маршрутизаторов в Интернете может быть один маршрут для всех адресов в блоке, позволяющий избежать необходимости иметь маршруты для всех меньших частей этого блока. Например, на рис. 26.4 представлены три компании с тремя разными глобальными префиксами маршрутизации IPv6; у маршрутизатора справа (R4) есть один маршрут IPv6 для каждого глобального префикса маршрутизации.

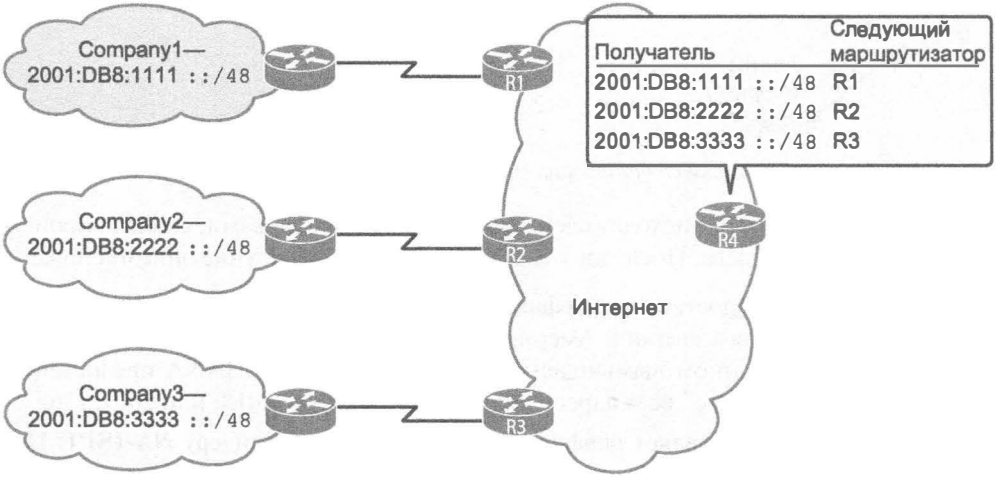


Рис. 26.4. Три глобальных префикса маршрутизации с одним маршрутом на префикс

Глобальный префикс маршрутизации выделяет IPv6-адреса для использования одной отдельной компанией, точно так же, как открытая сеть IPv4 или блок адресов CIDR в протоколе IPv4. Все IPv6-адреса в компании должны начинаться с того же глобального префикса маршрутизации, во избежание использования IPv6-адресов других компаний. Никакие другие компании не должны использовать IPv6-адреса с этим префиксом. К счастью, протокол IPv6 предоставляет пока еще много пространства и позволяет всем компаниям иметь глобальный префикс маршрутизации с большим количеством адресов.

Процесс присвоения как IPv6-, так и IPv4-адресов полагается на те же организации: IANA (наряду с ICANN), региональный реестр Интернета (Regional Internet Registry — RIR) и провайдеры услуг Интернета. Предположим, например, что компания Company1 получила глобальный префикс маршрутизации. Префикс 2001:0DB8:1111::/48 означает "все адреса с первыми 12 шестнадцатеричными цифрами 2001:0DB8:1111". Чтобы получить его, осуществляется процесс, представленный на рис. 26.5.

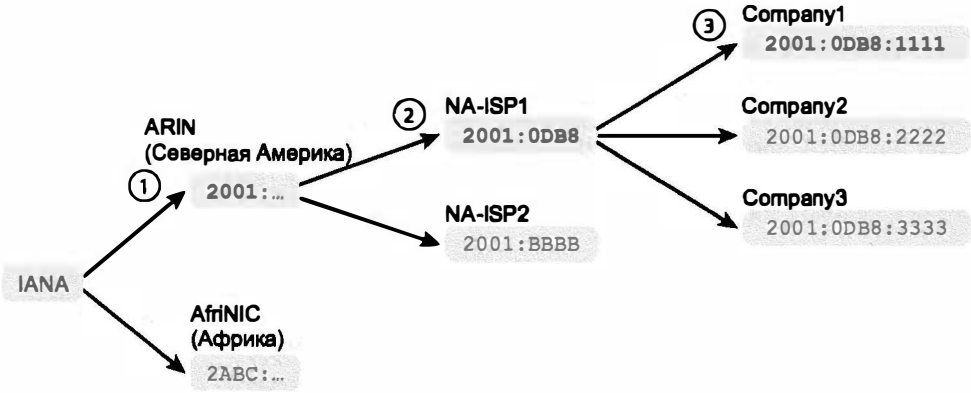


Рис. 26.5. Присвоение префикса организациями IANA, RIR и провайдерами услуг Интернета

События на рисунке происходят слева направо; другими словами, сначала происходит событие, крайнее слева. Последовательность событий на рисунке описана ниже.

1. **Агентство IANA предоставляет префикс ARIN 2001::/16**. Реестр ARIN (региональный реестр RIR для Северной Америки) запрашивает у агентства IANA большой блок адресов. В данном вымышленном примере агентство IANA предоставляет реестру ARIN префикс "всех адресов, начинающихся на 2001", или 2001::/16.
2. **Реестр ARIN предоставляет префикс 2001:0DB8::/32 провайдеру NA-ISP1**. Провайдер NA-ISP1, вымышленный провайдер в Северной Америке, запрашивает у реестра ARIN новый префикс IPv6. Реестр ARIN получает подмножество префиксов 2001::/16, а именно все адреса, начинающиеся с 32 битов (8 шестнадцатеричных цифр), 2001:0DB8, и предоставляет его ISP.
3. **Провайдер NA-ISP1 предоставляет префикс 2002:0DB8:1111::/48 компании Company1**. Компания Company1 принимает решение о переходе на IPv6-адреса, поэтому она обращается к своему провайдеру NA-ISP1 за блоком глобальных одноадресатных адресов. Провайдер NA-ISP1 предоставляет компании Company1 малую часть своего блока адресов, в данном случае адреса, которые начинаются с 48 битов (12 шестнадцатеричных цифр), 2001:0DB8:1111 (2001:0DB8:1111::/48).

ВНИМАНИЕ!

Если подключение к Интернету по протоколу IPv6 не планируется, то некоторое время (только для экспериментов) можно не запрашивать глобальный префикс маршрутизации IPv6. Просто возьмите IPv6-адреса и настройте свои устройства.

Адресные интервалы для глобальных одноадресатных адресов

Глобальные одноадресатные адреса составляют большую часть пространства IPv6-адресов. Однако, в отличие от протокола IPv4, правила разделения IPv6-адресов на категории существенно гибче, чем правила для классов A, B, C, D и E протокола IPv4.

Первоначально агентство IANA резервировало все IPv6-адреса, начинающиеся с шестнадцатеричных 2 или 3, как глобальные одноадресатные. (Как префикс этот адресный интервал может быть записан кратко так: 2000::/3.)

Более поздние запросы на комментарии расширили глобальный диапазон одноадресатных адресов до всех IPv6-адресов, не зарезервированных для использования в других целях. Например, все обсуждаемые далее в этой главе уникальные локальные одноадресатные адреса начинаются с шестнадцатеричных цифр FD. Хотя глобальные одноадресатные адреса не включают начинающиеся с FD, любые не зарезервированные специально адресные интервалы считаются глобальными одноадресатными.

И наконец, только потому, что столь огромное количество адресов находится в пределах диапазона глобально одноадресатных, агентство IANA не присваивает префиксы из всего адресного интервала. По общему признанию, протокол IPv4 отлично выживал более тридцати лет со значительно меньшим количеством адресов. При взвешенном и осмысленном подходе к назначению IPv6-адресов это пространство может продержаться даже дольше, чем пространство IPv4.

Список обсуждаемых в этой книге префиксов адресов и их задач приведен в табл. 26.1.

Таблица 26.1. Некоторые из типов IPv6-адресов и их первые шестнадцатеричные цифры

Ключевая тема

Тип адреса	Первые шестнадцатеричные цифры
Глобальный одноадресатный	2 или 3 (первоначально); все не зарезервированные на настоящий момент
Уникальный локальный	FD
Многоадресатный	FF
Локальный канала связи	FE80

Создание подсетей IPv6 с использованием глобальных одноадресатных адресов

После получения предприятием блока глобальных одноадресатных адресов, другими словами, глобального префикса маршрутизации, компания должна разделить этот большой блок адресов на подсети.

Подсети IPv6 создаются главным образом как и подсети IPv4, но математический механизм у них проще. Благодаря такому большому количеству доступных адресов обычно используют самую простую длину префикса IPv6: /64. Использование длины префикса /64 для всех подсетей делает математический механизм создания подсетей IPv6 столь же простым, как использование маски /24 для всех подсетей IPv4. Кроме того, процесс динамического присвоения IPv6-адресов лучше работает с длиной префикса /64. Практические задания этой книги предполагают, что проекты IPv6 будут использовать длину префикса /64.

В настоящем разделе рассматриваются различные этапы создания подсетей IPv6, главным образом на примерах, использующих длину префикса /64. В разделе приводятся правила, согласно которым одни адреса должны быть в одной подсети, а другие в других подсетях. Кроме того, в этом разделе рассматривается, как проанализировать глобальный префикс маршрутизации и связанную с ним длину пре-

фикса, чтобы найти все префиксы IPv6 (идентификаторы подсети) и адреса в каждой подсети.

ВНИМАНИЕ!

Если концепции создания подсетей IPv4 понятны не до конца, еще раз перечитайте главу 11, в которой они рассматриваются.

Решение о необходимости подсетей IPv6

В первую очередь, протоколы IPv6 и IPv4 используют те же концепции о необходимости подсетей: по одной для каждой сети VLAN и по одной для каждого двухточечного соединения WAN (последовательного или EoMPLS). Подробности создания подсетей для Frame Relay обсуждаются в книге по ICND2. На рис. 26.6 приведен пример концепции использования объединенной сети малого предприятия Company1. У предприятия Company1 две локальные сети с двухточечным последовательным каналом связи, соединяющим площадки. У нее есть также канал связи Ethernet WAN, соединенный с провайдером ISP. По той же логике, что и у протокола IPv4, компании Company1 требуется четыре подсети IPv6.

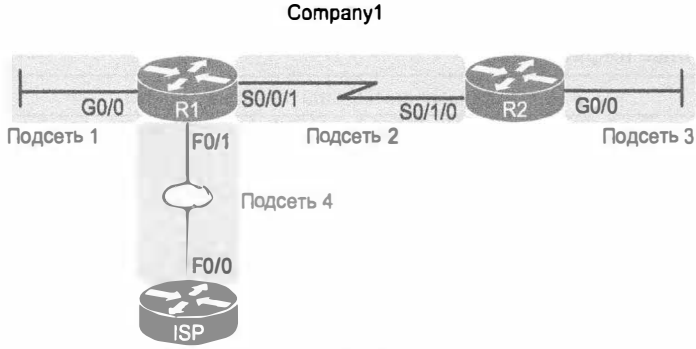


Рис. 26.6. Области для подсетей IPv6

Механика создания подсетей IPv6 и глобальные одноадресатные адреса

Чтобы понять, как разделить на подсети большой блок IPv6-адресов, необходимо знать теорию и механизм использования протокола IPv6. Чтобы помочь изучить эти детали, можно сравнить некоторые концепции протокола IPv6 с подобными понятиями протокола IPv4.

До создания подсетей у IPv4-адреса есть две части: часть сети и часть хоста. Правила классов A, B и C определяют длину части сети и части хоста, составляющей остальную часть 32-разрядного IPv4-адреса, как показано на рис. 26.7.

Разделяя на подсети сеть IPv4 класса A, B или C, сетевой инженер предприятия принимает некие решения. Концептуально он создает трехчастное представление адресов, добавляя в центр поле подсети за счет сокращения поля хоста. (Многие называют их "битами, позаимствованными у хоста".) Размер части сети неизменен, он обусловлен правилами класса A, B и C, а линия между частями подсети и хоста перемещается на основании выбранной маски подсети. На рис. 26.8 представлена концепция разделения адреса сети класса B.

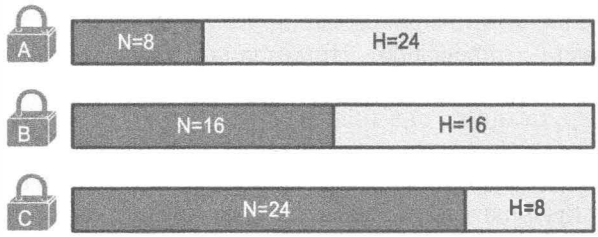


Рис. 26.7. Представление не разделенных на подсети классовых IPv4-адресов

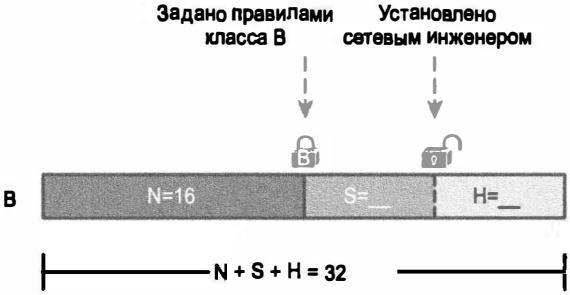


Рис. 26.8. Представление IPv4-адреса разделенной на подсети классовой сети

Протокол IPv6 использует концепцию, подобную представленной на рис. 26.9. Структура демонстрирует три главных части, начиная с глобального префикса маршрутизации, являющегося исходным значением, совпадающим у всех IPv6-адресов предприятия. Адрес заканчивается идентификатором интерфейса, аналогичным полю хоста IPv4. Поле подсети находится между этими двумя полями, используемыми как числовой путь и идентификатор подсети, очень похожий на поле подсети в IPv4-адресах.

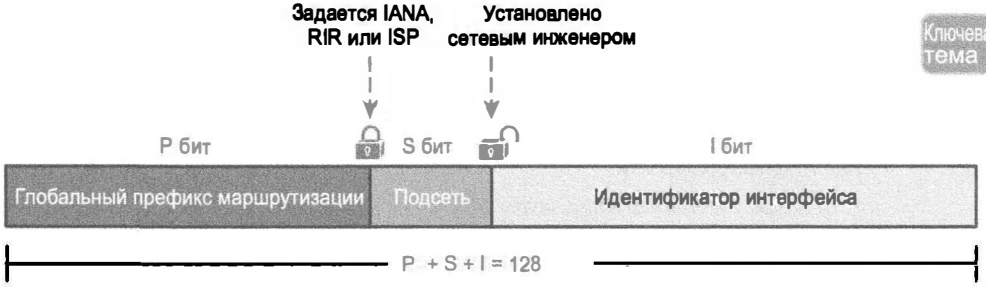


Рис. 26.9. Структура глобального одноадресатного IPv6-адреса после разделения на подсети

Сначала сравните общее представление на рис. 26.8 и рис. 26.9. Глобальный префикс маршрутизации IPv6 действует как часть сети в структуре IPv4-адреса. Часть подсети IPv6 действует как и часть подсети IPv4. Правая сторона IPv6-адреса формально называется идентификатором интерфейса, он действует как часть хоста IPv4-адреса.

Теперь сосредоточимся на глобальном префиксе маршрутизации IPv6 и его длине. В отличие от протокола IPv4, у протокола IPv6 нет концепции классов адресов,

поэтому длину глобального префикса маршрутизации никакие предварительно установленные правила не определяют. Но когда компания обращается к провайдеру ISP, RIR или любой другой организации, имеющей право предоставлять глобальный префикс маршрутизации, она получает и префикс, и длину префикса. Полученная компанией длина префикса, как правило, фиксируется и больше не изменяется на протяжении довольно продолжительного времени. (Обратите внимание, что длина глобального префикса маршрутизации, как правило, находится в диапазоне от /32 до /48, но пока возможно и /56.)

Затем обратимся к полю идентификатора интерфейса (см. рис. 26.9, *справа*). По некоторым причинам, которые станут более очевидными впоследствии, это поле зачастую имеет длину 64 бита. Должно ли оно быть длиной 64 бита? Нет. Но 64-битовые идентификаторы интерфейса хорошо работают в реальных сетях, и нет никаких причин избегать использования полей идентификатора интерфейса длиной 64 бита.

И наконец, обратимся к полю подсети в центре рис. 26.9. Как и в протоколе IPv4, это поле является местом для номера подсети IPv6. Длину поля подсети определяют два аспекта: длина глобального префикса маршрутизации и длина идентификатора интерфейса. При общепринятом 64-битовом поле идентификатора интерфейса поле подсети обычно составляет 64 – P битов, где P — длина глобального префикса маршрутизации.

Теперь рассмотрим структуру конкретного глобального одноадресатного IPv6-адреса 2001:0DB8:1111:0001:0000:0000:0000:0001, представленного на рис. 26.10. В данном случае:

- компании был присвоен префикс 2001:0DB8:1111 с длиной префикса /48;
- компания использует обычный 64-битовый идентификатор интерфейса;
- в компании принято 16 битов поля подсети, что допускает 2^{16} подсетей IPv6.

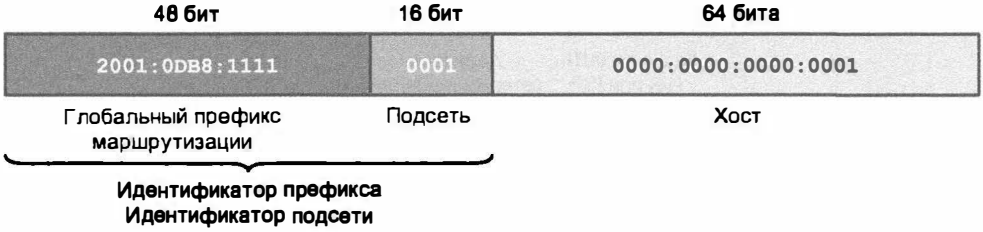


Рис. 26.10. Пример структуры адреса для компании Company1

Наряду с математическими аспектами пример на рис. 26.10 демонстрирует одну из причин использования большинством компаний длины префикса /64 для всех подсетей. В соответствии с этой структурой компания Company1 способна поддерживать 2^{16} подсетей (65 536). Немногим компаниям нужно так много подсетей. Каждая подсеть способна содержать более чем 10^{18} адресов (2^{64} минус несколько зарезервированных номеров). Таким образом, и для подсетей, и для хостов в структуре адреса намного больше места, чем необходимо. Кроме того, длина префикса /64 для всех подсетей существенно упрощает математику, поскольку она разделяет 128-битовый IPv6-адрес строго пополам.

Все идентификаторы подсети IPv6

Как и протокол IPv4, протокол IPv6 должен идентифицировать каждую подсеть IPv6 некоторым идентификатором — идентификатором подсети. На рис. 26.10 приведены неофициальное название этого числа (идентификатор подсети) и официальное (идентификатор префикса). Маршрутизаторы содержат идентификатор подсети IPv6 в таблицах маршрутизации наряду с длиной префикса.

В главе 25 уже обсуждалось, как найти идентификатор подсети по заданному IPv6-адресу и длине префикса. Математика та же, что и у глобальных одноадресатных, а также уникальных локальных адресов, обсуждаемых далее в главе. Поскольку математический механизм уже изложен в главе 25, не будем его повторять. Но для завершенности скажем, что идентификатором подсети в примере на рис. 26.10 был бы 2001:DB8:1111:1::/64.

Все подсети IPv6

Решение использовать во всех подсетях IPv4 единую маску позволяет находить и записывать все подсети сети класса A, B или C, используя единую маску подсети. В протоколе IPv6 применены те же идеи. Если планируется использовать единую длину префикса для всех подсетей, можно также начать с глобального префикса маршрутизации и записать все идентификаторы подсети IPv6.

Для поиска всех идентификаторов подсети следует найти все уникальные значения, соответствующие части подсети IPv6-адреса. Для этого достаточно придерживаться следующих правил.

Правила поиска всех идентификаторов подсети IPv6 по за данному глобальному префиксу маршрутизации и одинаковой длине префикса для всех подсетей

Ключевая тема

- Все идентификаторы подсети начинаются с глобального префикса маршрутизации.
- Для идентификации каждой отдельной подсети используйте отличное значение в поле подсети.
- В части идентификатора интерфейса все идентификаторы подсети содержат нули.

Для примера возьмем проект IPv6, представленный на рис. 26.10, и подумаем обо всех идентификаторах подсети. В первую очередь, все подсети будут использовать общепринятую длину префикса /64. Данная компания использует глобальный префикс маршрутизации 2001:0DB8:1111::/48, определяющий первые 12 шестнадцатеричных цифр всех идентификаторов подсети. Для поиска всех возможных идентификаторов подсети IPv6 запишите все комбинации уникальных значений в четвертом квартете, а затем представьте все четыре последних квартета, заполненных нулями, символом ::. На рис. 26.11 приведено только начало такого списка.

Пример допускает 65 536 подсетей, поэтому, безусловно, приведены не все возможные подсети. Но все комбинации шестнадцатеричных значений в четвертом квартете приведены.

	2001:0DB8:1111:0000::	2001:0DB8:1111:000B::
✓	2001:0DB8:1111:0001::	2001:0DB8:1111:000C::
✓	2001:0DB8:1111:0002::	2001:0DB8:1111:000D::
✓	2001:0DB8:1111:0003::	2001:0DB8:1111:000E::
✓	2001:0DB8:1111:0004::	2001:0DB8:1111:000F::
	2001:0DB8:1111:0005::	2001:0DB8:1111:0010::
	2001:0DB8:1111:0006::	2001:0DB8:1111:0011::
	2001:0DB8:1111:0007::	2001:0DB8:1111:0012::
	2001:0DB8:1111:0008::	2001:0DB8:1111:0013::
	2001:0DB8:1111:0009::	2001:0DB8:1111:0014::
	2001:0DB8:1111:000A::	2001:0DB8:1111:0015::
	Глобальный префикс маршрутизации / Подсеть	Глобальный префикс маршрутизации / Подсеть

Рис. 26.11. 22 первые возможные подсети с 16-разрядным полем подсети (в данном случае)

ВНИМАНИЕ!

Более формально идентификатор подсети IPv6 называется *альтернативным адресом маршрутизатора подсети* (subnet router anycast address). Он зарезервирован и не должен использоваться как IPv6-адрес ни для какого хоста.

Применение подсетей в топологии объединенной сети

После составления инженером перечня всех возможных идентификаторов подсети (на основании проекта подсетей) следующий шаг подразумевает выбор идентификаторов подсети, используемых для каждого канала связи, нуждающегося в подсети IPv6. Как и в случае IPv4, в подсетях IPv6 нуждаются все сети VLAN, все последовательные каналы связи, все каналы связи EoMPLS и многие другие каналы связи.

В примере на рис. 26.12 снова показана компания Company1. Здесь использованы четыре подсети, приведенные на рис. 26.11, с обозначениями около них. Обозначения — это только напоминание, чтобы не использовать их в других местах.

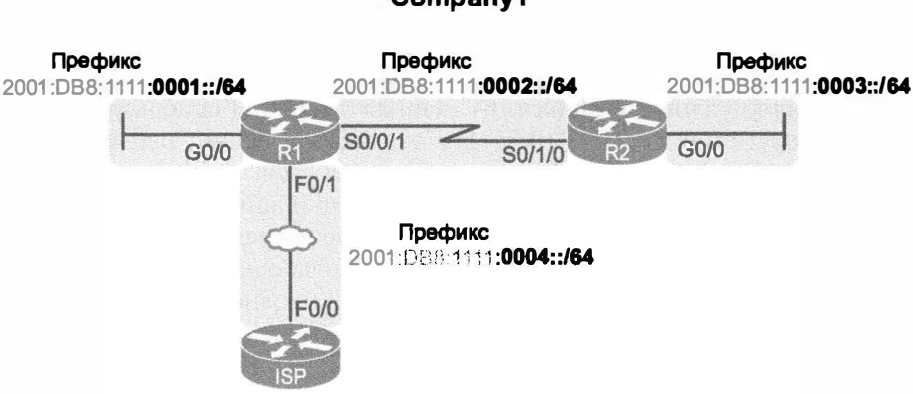


Рис. 26.12. Подсети компании Company1 с глобальным префиксом маршрутизации 2001:0DB8:1111::/48

Присвоение адресов хостам в подсети

Теперь, когда инженер распланировал, какая подсеть IPv6 будет использоваться в каждой области, можно планировать и реализовать индивидуальные IPv6-адреса. Каждый адрес должен быть уникальным, ни один другой интерфейс хоста не должен использовать тот же IPv6-адрес. Кроме того, хосты не могут использовать сам идентификатор подсети.

Процесс назначения IPv6-адресов интерфейсам протекает таким же образом, как и в протоколе IPv4. Адреса могут быть заданы статически наряду с длиной префикса, а также IPv6-адресами стандартного маршрутизатора и сервера DNS. Те же параметры могут быть изучены хостами динамически с использованием протокола DHCP или другого встроенного механизма IPv6 — *автоматической настройки адреса* (Stateless Address Autoconfiguration — SLAAC).

На рис. 26.13 приведен пример нескольких статических IP-адресов, которые могли быть присвоены интерфейсам маршрутизатора на основании выбора подсетей, представленных на рис. 26.12. В каждом случае интерфейсы маршрутизатора используют идентификатор интерфейса, являющийся относительно малым числом, которое легко запомнить.

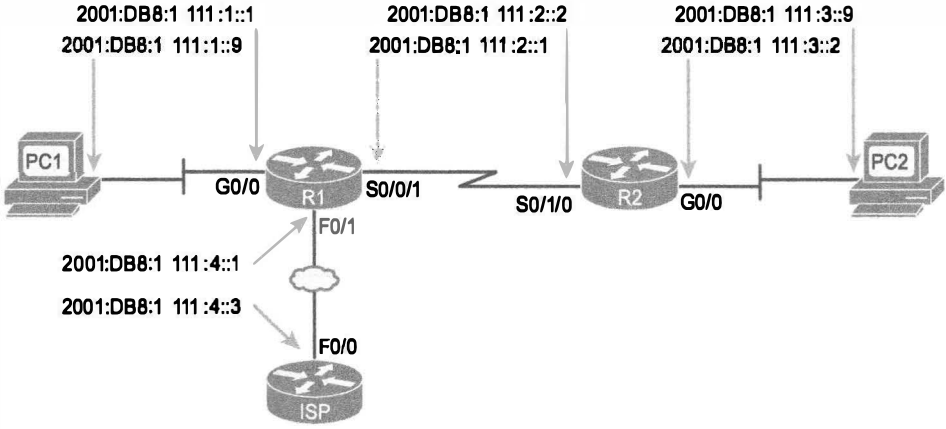


Рис. 26.13. Пример статических IPv6-адресов, назначенных на основании проекта подсетей на рис. 26.12

Отложим пока детали настройки IPv6-адреса до следующих двух глав. В главе 27 рассматривается как статическая, так и динамическая настройка IPv6-адресов на маршрутизаторах. В главе 28 описана настройка IPv6-адресов на хостах и больше внимания уделяется динамическим методам и связанным с ними протоколам.

Уникальные локальные одноадресатные адреса

Уникальные локальные одноадресатные адреса действуют как частные IPv6-адреса. У этих адресов много сходств с глобальными одноадресатными адресами, особенно в отношении подсетей. Наибольшее отличие заключается в литерале числа (уникальные локальные адреса начинаются с шестнадцатеричного FD) и процессе администрирования: уникальные локальные префиксы не регистрируются ни в каких инстанциях и доступны для применения многими организациями.

Хотя сетевой инженер создает уникальные локальные адреса без всякой регистрации, процесс присвоения адресов все еще должен подчиняться некоторым правилам.

Ключевая тема

Правила создания уникальных локальных одноадресатных адресов

- Используйте две первые шестнадцатеричные цифры FD.
- Выберите уникальный 40-битовый глобальный идентификатор.
- Добавьте к "FD" глобальный идентификатор, чтобы получить 48-битовый префикс, используемый для всех адресов.
- Используйте следующие 16 битов как поле подсети.
- Обратите внимание: структура оставляет на поле идентификатора интерфейса как раз 64 бита.

Формат этих уникальных локальных одноадресатных адресов приведен на рис. 26.14.

Ключевая тема

Рис. 26.14. Формат уникального локального одноадресатного IPv6-адреса

ВНИМАНИЕ!

Только для справки: фактически агентство IANA резервирует для этих адресов префиксы FC00::/7, а не FD00::/8. Префикс FC00::/7 включает все адреса, начинающиеся с шестнадцатеричного FC и FD. Но документ RFC (4193) требует, чтобы 8-й бит этих адресов был установлен в 1, поэтому сейчас практически все уникальные локальные адреса начинаются с первых двух цифр FD.

Создание подсетей с уникальными локальными IPv6-адресами

Создание подсетей с использованием уникальных локальных адресов осуществляется точно так же, как и создание подсетей с глобальными одноадресатными адресами и 48-битовым глобальным префиксом маршрутизации. Единственное различие в том, что с глобальными одноадресатными адресами процесс начинается с запроса компанией глобального префикса маршрутизации, который мог бы иметь (а мог бы и не иметь) длину префикса /48. При уникальных локальных адресах этот префикс создается локально и начинается с /48 при первом наборе из 8 битов и следующих случайно выбранных 40 битов.

Процесс может быть таким же простым, как выбор 40-битового значения глобального идентификатора. 40 битов требуют 10 шестнадцатеричных цифр, поэтому можно даже избежать двоичной математики и просто составить значение из десяти уникальных шестнадцатеричных цифр. Предположим, например, что выбран 40-битовый глобальный идентификатор 00 0001 0001. Адреса должны начинаться с двух шестнадцатеричных цифр FD, поэтому весь префикс будет FD00:0001:0001::/48 или FD00:1:1::/48 в сокращенном виде.

Для создания подсетей, как и в прежних примерах с 48-битовым глобальным префиксом маршрутизации, весь четвертый квартет рассматривается как поле подсети (см. рис. 26.14).

На рис. 26.15 приведен план подсетей примера, использующего уникальные локальные адреса. В примере повторяется та же самая топология, что и ранее на рис. 26.12; на этом рисунке представлены подсети с глобальным одноадресатным префиксом. Данный пример использует те же числа для поля подсети четвертого квартета, просто 48-битовый глобальный одноадресатный префикс заменен новым локальным уникальным префиксом FD00:1:1.

Рис. 26.15. Создание подсетей с использованием уникальных локальных адресов

Потребность в глобально уникальных локальных адресах

Пример на рис. 26.15 демонстрирует легко запоминающийся префикс FD00:1:1::/48. Ранее в этом примере был представлен легко запоминающийся глобальный идентификатор. Какой глобальный идентификатор вы выбрали бы для своей компании? Выбрали бы вы номер, который нельзя сократить и сделать короче? Если бы вы могли сами выбрать префикс IPv6 для уникальных локальных адресов из приведенных в следующем списке, то какой бы вы выбрали для своей компании?

- FDE9:81BE:A059::/48
- FDF0:E1D2:C3B4::/48
- FD00:1:1::/48

При свободе выбора большинство людей выбрали бы легко запоминающийся и короткий при вводе префикс FD00:1:1::/48. И в лабораторных условиях, и в небольшой проверочной сети выбор удобного номера вполне разумен. Но в реальных корпоративных сетях выбрать глобальный идентификатор по своему усмотрению не получится — придется следовать правилам уникальных локальных адресов, призванных помочь сделать адреса уникальными в своей области, даже если префикс не регистрирует ISP или RIR.

Уникальные локальные адреса определяет документ RFC 4193. Документ RFC подчеркивает важность выбора такого глобального идентификатора, который вряд ли

используется другими компаниями. Каков смысл уникальных глобальных идентификаторов в каждой компании? Создание всех уникальных локальных адресов в их индивидуальности в мировом масштабе. Так, если действительно планируется использовать уникальные локальные адреса в реальной сети, используйте для создания префикса логику генератора случайных чисел, описанную в документе RFC 4193.

Одна из главных причин попытки использования уникального префикса, а не самостоятельно выбранных легко запоминающихся префиксов заключается в готовности к тому дню, когда ваша компания окажется "слита" или куплена другой компанией. Сегодня, используя протокол IPv4, большой процент компаний использует частную сеть IPv4 10.0.0.0. Когда компании объединяют свои сети, факт использования обеими компаниями одинаковой сети 10.0.0.0 существенно затрудняет объединение сетей, по сравнению с использованием разных частных сетей IPv4. При использовании уникальных локальных IPv6-адресов, если обе компании поступили правильно и выбрали префикс случайным образом, то вероятней всего они используют разные префиксы, что намного упростит слияние сетей компаний. Однако компании, выбирающие легкий, на первый взгляд, способ применив легко запоминающейся префикс FD00:1:1, повышают риск дополнительных усилий при слиянии с другой компанией, которая также решила использовать тот же префикс.

Обзор

Резюме

- Для одноадресатной адресации протокол IPv6 предоставляет две подобные возможности.
  - Глобальный одноадресатный адрес. Аналог открытых IPv4-адресов. Организация, нуждающаяся в IPv6-адресах, запрашивает зарегистрированный блок IPv6-адресов, присваиваемый как глобальный префикс маршрутизации. После этого только данная организация использует адреса из этого блока, т.е. адреса, начинающиеся с присвоенного префикса.
  - Уникальный локальный адрес. Аналог частных IPv4-адресов. Несколько организаций вполне могут использовать эти же адреса без необходимости регистрировать их в соответствующих инстанциях.
- Этот зарезервированный блок IPv6-адресов может использовать только одна компания, поэтому он называется *глобальным префиксом маршрутизации*. Каждая организация, которая собирается подключиться к Интернету и использовать глобальные одноадресатные IPv6-адреса, должна запросить и получить глобальный префикс маршрутизации. В самом общем случае глобальный префикс маршрутизации можно считать номером сети класса A, B или C из диапазона открытых IPv4-адресов.
- Таким образом, ниже приведены некоторые из типов IPv6-адресов и их первые шестнадцатеричные цифры.

Тип адреса	Первые шестнадцатеричные цифры
Глобальный одноадресатный	2 или 3 (первоначально); все не зарезервированные на настоящий момент
Уникальный локальный	FD
Многоадресатный	FF
Локальный канала связи	FE80

- В первую очередь, оба протокола, IPv6 и IPv4, используют те же концепции о необходимости подсетей: по одной для каждой сети VLAN и по одной для каждого двухточечного соединения WAN (последовательного или EoMPLS).
- Протокол IPv6 использует концепцию, подобную представленной на рис. 26.9. Структура демонстрирует три главных части, начиная с глобального префикса маршрутизации, являющегося исходным значением, совпадающим у всех IPv6-адресов предприятия. Адрес заканчивается идентификатором интерфейса, аналогичным полю хоста IPv4. Поле подсети находится между этими двумя полями, используемыми как числовой путь и идентификатор подсети, очень похожий на поле подсети в IPv4-адресах.
- Для поиска всех идентификаторов подсети следует найти все уникальные значения, соответствующие части подсети IPv6-адреса. Для этого достаточно придерживаться следующих правил.

- Все идентификаторы подсети должны начинаться с глобального префикса маршрутизации.
- Для идентификации каждой отдельной подсети используйте отличное значение в поле подсети.
- В части идентификатора интерфейса все идентификаторы подсети должны содержать нули.

- Уникальные локальные одноадресатные адреса действуют как частные IPv6-адреса. У этих адресов много сходств с глобальными одноадресатными адресами, особенно в отношении подсетей. Наибольшее отличие заключается в литерале числа (уникальные локальные адреса начинаются с шестнадцатеричного FD) и процессе администрирования: уникальные локальные префиксы не регистрируются ни в каких инстанциях и доступны для применения многими организациями.
- Хотя сетевой инженер создает уникальные локальные адреса без всякой регистрации, процесс присвоения адресов все еще должен подчиняться некоторыми правилам.
  - Используйте две первые шестнадцатеричные цифры FD.
  - Выберите уникальный 40-битовый глобальный идентификатор.
  - Добавьте к "FD" глобальный идентификатор, чтобы получить 48-битовый префикс, используемый для всех адресов.
  - Используйте следующие 16 битов как поле подсети.
  - Обратите внимание: структура оставляет на поле идентификатора интерфейса как раз 64 бита.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Какой из следующих IPv6-адресов, на основании его первых шестнадцатеричных цифр, является уникальным локальным одноадресатным адресом?
   А) 3123:1:3:5::1
   Б) FE80::1234:56FF:FE78:9ABC
   В) FDAD::1
   Г) FF00::5
2. Какой из следуюших IPv6-адресов, на основании его первых шестнадцатеричных цифр, является глобальным одноадресатным адресом?
   А) 3123:1:3:5::1
   Б) FE80::1234:56FF:FE78:9ABC
   В) FDAD::1
   Г) FF00::5
3. При создании подсетей инженеру представлен IPv6-адрес, структура которого разделена на три части. Какая часть структуры IPv6-адреса концептуально больше всего походит на часть сети в трехчастной структуре IPv4-адреса?

А) Подсеть.

Б) Идентификатор интерфейса.

В) Сеть.

Г) Глобальный префикс маршрутизации.

Д) Альтернативный адрес маршрутизатора подсети

4. При создании подсетей инженеру представлен IPv6-адрес, структура которого разделена на три части. С учетом того, что все подсети используют ту же длину префикса, какой из следующих ответов содержит имя поля в крайней правой части адреса?

А) Подсеть.

Б) Идентификатор интерфейса.

В) Сеть.

Г) Глобальный префикс маршрутизации.

Д) Альтернативный адрес маршрутизатора подсети.

5. Какую часть IPv6-адреса FD00:1234:5678:9ABC:DEF1:2345:6789:ABCD считают глобальным идентификатором уникального локального адреса?

А) Ни один; у этого адреса нет глобального идентификатора.

Б) 00:1234:5678:9ABC

В) DEF1:2345:6789:ABCD

Г) 00:1234:5678

Д) FD00

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 26.2.

Таблица 26.2. Ключевые темы главы 26

Элемент	Описание	Страница
Список	Сетевые каналы связи, нуждающиеся в подсетях IPv6	756
Список	Два типа одноадресатных адресов IPv6	758
Табл. 26.1	Некоторые из типов IPv6-адресов и их первые шестнадцатеричные цифры	761
Рис. 26.9	Структура глобального одноадресатного IPv6-адреса после разделения на подсети	763
Список	Правила поиска всех идентификаторов подсети IPv6 по за данному глобальному префиксу маршрутизации и одинаковой длине префикса для всех подсетей	765
Список	Правила создания уникальных локальных одноадресатных адресов	768
Рис. 26.14	Формат уникального локального одноадресатного IPv6-адреса	768

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

глобальный одноадресатный адрес (global unicast address), глобальный префикс маршрутизации (global routing prefix), уникальный локальный адрес (unique local address), идентификатор подсети (subnet ID), идентификатор префикса (prefix ID), альтернативный адрес маршрутизатора подсети (subnet router anycast address)

Ответы на контрольные вопросы:

1 В. **2** А. **3** Г. **4** Б. **5** Г.

Реализация IPv6-адресации на маршрутизаторах

При IPv4-адресации некоторые устройства, такие как серверы и маршрутизаторы, как правило, используют статически заданные IPv4-адреса. Устройства конечного пользователя ничуть не страдают, если их адрес время от времени изменяется; обычно они получают его динамически, используя протокол DHCP. При протоколе IPv6 серверы, маршрутизаторы и другие устройства используют тот же общий режим, контролируемый группой технических специалистов, зачастую применяющих для них предопределенные IPv6-адреса, а устройства конечного пользователя, как правило, применяют динамически изученные IPv6-адреса.

Данная глава посвящена адресам, задаваемым на маршрутизаторах, а в главе 28 рассматриваются адреса, изучаемые хостами IPv6.

Маршрутизаторы требуют наличия на интерфейсах одноадресатных IPv6-адресов. В то же время для выполнения большинства обязанностей и ролей маршрутизаторам нужно множество других IPv6-адресов. Эта глава начинается с наиболее очевидного — настройки IPv6-адресации, соответствующей такому средству протокола IPv4, как настройка IPv6-адресов на интерфейсах и просмотр полученной конфигурации при помощи команды `show`. Затем речь пойдет о новых концепциях IPv6-адресации; будут также рассмотрены некоторые другие адреса, используемые маршрутизаторами при решении различных задач.

В этой главе рассматриваются следующие экзаменационные темы

IP-адресация (IPv4/IPv6)
- Технологические требования для запуска протокола IPv6 совместно с протоколом IPv4 как двойного стека.
- Описание IPv6-адреса.
  - Многоадресатный.
  - Локальный адрес канала связи.
  - Адрес в формате eui 64.

Технологии маршрутизации IP
- Настройка и проверка применения CLI для установки базовой конфигурации маршрутизатора.
  - Команды Cisco IOS для базовой настройки маршрутизатора.
- Настройка и проверка состояния интерфейса Ethernet.
- Проверка конфигурации маршрутизатора и сетевого подключения.
  - Команды Cisco IOS для просмотра базовой информации маршрутизатора и сетевого подключения.

Поиск и устранение неисправностей
- Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Реализация одноадресатных IPv6-адресов на маршрутизаторах

Каждая компания базирует свою корпоративную сеть на одной или нескольких моделях или стеках протоколов. На ранних этапах существования корпоративные сети использовали один или несколько стеков протоколов от разных производителей (рис. 27.1, *слева*). Спустя довольно продолжительное время, компании добавили в этот набор стандарт TCP/IP (на основании протокола IPv4). В конечном счете компании полностью перешли на стандарт TCP/IP как единственный стек используемых протоколов.

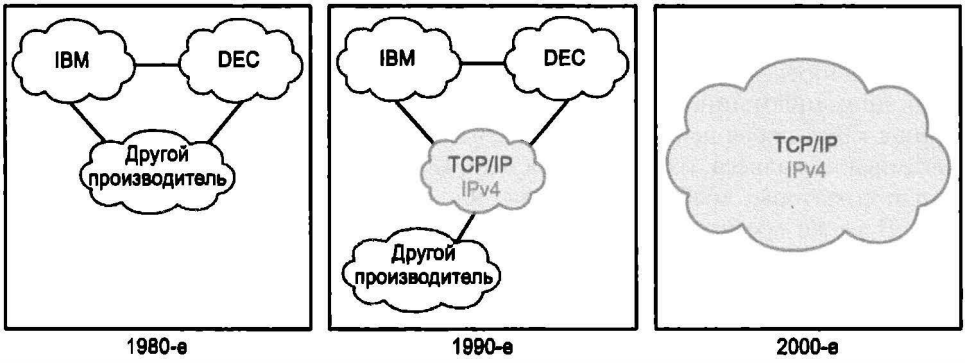


Рис. 27.1. Переход корпоративных сетей на использование только протокола TCP/IP

Появление протокола IPv6 требует его реализации на хостах конечного пользователя, серверах, маршрутизаторах и других устройствах. Но корпорации не могут перевести все устройства с протокола IPv4 на IPv6 за один день. Вероятней всего, понадобится некоторый довольно продолжительный переходный период (возможно, несколько лет), на протяжении которого корпоративная сеть снова будет использовать несколько стеков протоколов: один на основании IPv4 и один на основании IPv6.

Ключевая тема

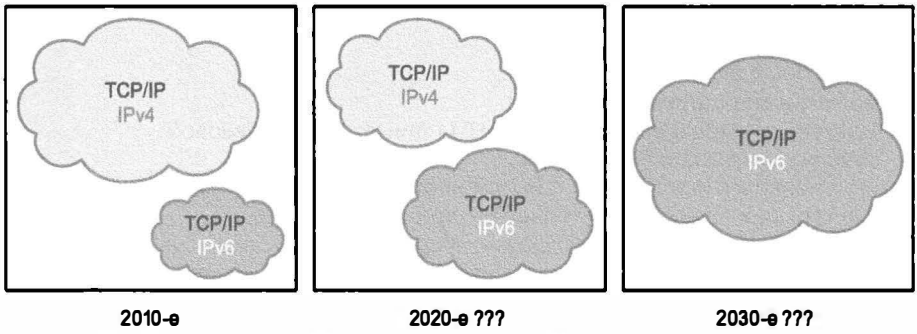


Рис. 27.2. Вероятный способ использования двух стеков (IPv4 и IPv6) на протяжении переходного периода

Один из способов поддержки протокола IPv6 в существующей объединенной корпоративной сети на базе протокола IPv4 заключается в реализации стратегии двойного стека. Для этого маршрутизаторы могут быть настроены на передачу пакетов IPv6 с IPv6-адресами на своих интерфейсах. Это похоже на то, как маршрутизаторы поддерживают протокол IPv4. Затем, по мере готовности, протокол IPv6 можно реализовать на хостах, используя и IPv4, и IPv6 (двойной стек). Первый раздел этой главы посвящен настройке и проверке одноадресатных IPv6-адресов на маршрутизаторах.

Статическая настройка одноадресатного адреса

Маршрутизаторы Cisco предоставляют две возможности для статической настройки IPv6-адресов. В одном случае задается полный 128-битовый адрес, в другом — лишь 64-битовый префикс, а вторую половину адреса (идентификатор интерфейса) маршрутизатор получает сам. Ниже представлены обе возможности, а также способ выбора маршрутизатором второй половины IPv6-адреса.

Настройка полного 128-битового адреса

Для статического задания полного 128-битового одноадресатного адреса (или глобального одноадресатного, или уникального локального адреса) маршрутизатор нуждается в подкоманде интерфейса `ipv6 address` *адрес/длина_префикса* на каждом интерфейсе. Адрес может быть сокращенным IPv6-адресом или полным шестнадцатеричным адресом с 32 цифрами. Команда включает значение длины префикса, причем между адресом и длиной префикса нет пробела.

Настройка IPv6-адреса интерфейса маршрутизатора действительно проста. Рис. 27.3 наряду с примерами 27.1 и 27.2 демонстрирует простой случай. Здесь представлен глобальный одноадресатный IPv6-адрес, используемый двумя разными маршрутизаторами на двух интерфейсах каждый. Как обычно, все подсети используют длину префикса /64.

Рис. 27.3. 128-битовые IPv6-адреса, заданные на интерфейсах маршрутизатора Cisco

Пример 27.1. Настройка статических IPv6-адресов на маршрутизаторе R1

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0
  ipv6 address 2001:DB8:1111:1::1/64
!
interface Serial0/0/0
  ipv6 address 2001:0db8:1111:0002:0000:0000:0000:0001/64
```

Пример 27.2. Настройка статических IPv6-адресов на маршрутизаторе R2

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0
  ipv6 address 2001:DB8:1111:3::2/64
!
interface Serial0/0/1
  ipv6 address 2001:db8:1111:2::2/64
```

ВНИМАНИЕ!

При настройке маршрутизатора R1 в примере 27.1 использовались и сокращенные, и несокращенные адреса, а также нижний и верхний регистр шестнадцатеричных цифр, демонстрируя допустимость их применения. Команды `show` маршрутизатора выводят сокращенное значение в верхнем регистре.

Разрешение маршрутизации IPv6

Хотя представленная в примерах 27.1 и 27.2 конфигурация сосредоточена в основном на настройке IPv6-адресов, они демонстрируют также важный, но зачастую пропускаемый этап настройки IPv6-адресов на маршрутизаторах Cisco — необходимость разрешить маршрутизацию IPv6.

Прежде чем маршрутизаторы смогут перенаправлять (маршрутизировать) пакеты IPv6, это следует сделать возможным. На маршрутизаторах Cisco маршрутизация IPv4 разрешена изначально, а маршрутизация IPv6 — нет. Решением является единственная команда — (`ipv6 unicast-routing`), разрешающая маршрутизацию IPv6 на маршрутизаторе.

Обратите внимание, что маршрутизация IPv6 должна быть разрешена на маршрутизаторе как глобально (`ipv6 unicast-routing`), так и на интерфейсе (`ipv6 address`), прежде чем он попытается перенаправить пакеты через этот интерфейс. (Если на маршрутизаторе случайно пропущена команда `ipv6 unicast-routing`, его интерфейсы могут быть полностью настроены IPv6-адресами, но маршрутизатор действует как хост IPv6 и не перенаправляет пакеты IPv6.)

Проверка настроенных IPv6-адресов

Протокол IPv6 использует множество команд `show`, подражающих синтаксису команд `show` протокола IPv4. Например:

- Команда `show ipv6 interface brief` предоставляет информацию об IPv6-адресе интерфейса, но не о длине префикса, как подобная команда `show ip interface brief` протокола IPv4.
- Команда `show ipv6 interface` предоставляет подробности о параметрах IPv6 интерфейса, что очень похоже на команду `show ip interface` протокола IPv4.

Одно из наиболее существенных отличий в общих командах заключается в том, что команда `show interfaces` все еще перечисляет IPv4-адреса и маски, но ничего не говорит о параметрах IPv6. Таким образом, чтобы просмотреть IPv6-адреса интерфейса, используйте команды, начинающиеся на `show ipv6`. В примере 27.3 приведено несколько команд для маршрутизатора R1 с объяснениями.

Пример 27.3. Проверка статических IPv6-адресов на маршрутизаторе R1

```
! Первый интерфейс находится в подсети 1
R1# show ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1FF:FE01:101
  No Virtual link-local address(es):
  Description: LAN at Site 1
  Global unicast address(es):
    2001:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::A
    FF02::1:FF00:1
    FF02::1:FF01:101
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.

R1# show ipv6 interface S0/0/0
Serial0/0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1FF:FE01:101
  No Virtual link-local address(es):
  Description: link to R2
  Global unicast address(es):
    2001:DB8:1111:2::1, subnet is 2001:DB8:1111:2::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::A
    FF02::1:FF00:1
    FF02::1:FF01:101
  MTU is 1500 bytes
! Строки пропущены для краткости

R1# show ipv6 interface brief
GigabitEthernet0/0          [up/up]
    FE80::1FF:FE01:101
    2001:DB8:1111:1::1
GigabitEthernet0/1          [administratively down/down]
    unassigned
Serial0/0/0                 [up/up]
    FE80::1FF:FE01:101
    2001:DB8:1111:2::1
Serial0/0/1                 [administratively down/down]
    unassigned
```

Сначала сосредоточимся на выводе двух команд `show ipv6 interface`, составляющем большую часть вывода в примере 27.3. Первая команда относится только к интерфейсу G0/0. Обратите внимание, что вывод перечисляет заданный IPv6-адрес, длину префикса и подсеть IPv6 (2001:DB8:1111:1::/64), вычисляемую маршрутизатором на основании IPv6-адреса. Вторая команда `show ipv6 interface` демонстрирует подобные подробности для интерфейса S0/0/0, часть которых пропущена.

В конце примера представлен вывод команды `show ipv6 interface brief`. Подобно команде IPv4 `show ip interface brief`, эта команда выводит IPv6-адреса, но не префиксы или их длины. Эта команда перечисляет также все интерфейсы на маршрутизаторе, а также разрешен ли на них протокол IPv6. В данном случае, например, единственными двумя интерфейсами маршрутизатора R1, обладающими IPv6-адресом, являются G0/0 и S0/0/0, как было указано ранее в примере 27.1.

Кроме IPv6-адресов на интерфейсах, маршрутизатор добавляет также в таблицу маршрутизации IPv6 подключенные маршруты IPv6 от каждого интерфейса. Как и в случае с протоколом IPv4, маршрутизатор сохраняет эти подключенные маршруты в таблице маршрутизации IPv6, только когда интерфейс находится в рабочем состоянии (up/up). Но если на интерфейсе настроен одноадресатный IPv6-адрес и интерфейс работает, то маршрутизатор добавляет подключенные маршруты. Пример 27.4 демонстрирует подключенный маршрут IPv6 на маршрутизаторе R1 согласно рис. 27.3.

Пример 27.4. Отображение подключенных маршрутов IPv6 на маршрутизаторе R1

```
R1# show ipv6 route connected
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix
       DCE - Destination, NDr - Redirect, O - OSPF Intra
       OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C      2001:DB8:1111:1::/64 [0/0]
       via GigabitEthernet0/0, directly connected
C      2001:DB8:1111:2::/64 [0/0]
       via Serial0/0/0, directly connected
```

Создание уникального идентификатора интерфейса с использованием EUI-64

Протокол IPv6 следует той же общей модели, что и протокол IPv4, согласно которой одни устройства обычно используют статические, предопределенные адреса, а другие — изученные динамически. Например, маршрутизаторы на предприятии, как правило, используют статические IPv4-адреса, а устройства конечного пользователя обычно изучают свои IPv4-адреса, используя протокол DHCP. При протоколе IPv6 маршрутизаторы также обычно используют статические IPv6-адреса, а пользовательские устройства используют протокол DHCP или SLAAC (Stateless Address Auto Configuration — *автоматическая настройка адреса без фиксации состояния*) для динамического изучения IPv6-адресов.

У маршрутизаторов есть две возможности для настройки стабильного, предсказуемого и неизменного IPv6-адреса интерфейса. Первый, уже упоминавшийся в этой

главе метод подразумевает использование команды `ipv6 address` для определения всего 128-битового адреса, как показано в примерах 27.1 и 27.2. Второй метод использует ту же команду `ipv6 address`, но для настройки только 64-битового префикса IPv6 интерфейса и позволяет маршрутизатору автоматически создать уникальный идентификатор интерфейса.

Второй метод подразумевает использование идентификатора EUI-64 (Extended Unique Identifier — *расширенный уникальный идентификатор*). Конфигурация включает ключевое слово, указывающее маршрутизатору использовать правила EUI-64 наряду с 64-битовым префиксом. Далее маршрутизатор использует правила EUI-64 для создания части идентификатора интерфейса в адресе следующим образом.

Правила создания IPv6-адреса с использованием правил EUI-64

Ключевая тема

1. Разделите 6-байтовый (12 шестнадцатеричных цифр) MAC-адрес на две половины (по 6 шестнадцатеричных цифр каждая).
2. Между этими двумя половинами вставьте часть FFFE, чтобы идентификатор интерфейса имел теперь в общей сложности 16 шестнадцатеричных цифр (64 бита).
3. Инвертируйте седьмой бит идентификатора интерфейса.

Главные элементы формирования такого адреса приведены на рис. 27.4.

Ключевая тема

Рис. 27.4. Формат IPv6-адреса с идентификатором интерфейса и частью EUI-64

Хотя это и кажется немного замысловатым, но работает. Кроме того, после небольшой практики можно быстро осмотреть IPv6-адрес и, обнаружив часть FFFE в середине идентификатора интерфейса, легко найти две половины MAC-адреса соответствующего интерфейса. Но чтобы предсказать IPv6-адрес на интерфейсе (в данном случае в формате EUI-64), следует быть готовым выполнить те же математические действия.

Например, если проигнорировать заключительный этап инвертирования седьмого бита, остальные этапы потребуют только перемещения частей. На рис. 27.5 представлены два примера этого процесса.

В обоих примерах показан тот же процесс. Каждый начинается с MAC-адреса, разделяемого на две половины (этап 2). На третьем этапе в середину вставляется часть FFFE, а на четвертом этапе четыре шестнадцатеричных цифры разделяются двоеточиями согласно соглашениям протокола IPv6.

Несмотря на то что примеры на рис. 27.5 представляют большинство этапов, заключительный этап пропущен. Этот этап требует преобразования первого байта (сначала две шестнадцатеричные цифры) из шестнадцатеричного в двоичный фор-

мат, инверсию седьмого из 8 битов и преобразование битов назад в шестнадцатеричный формат. Инверсия бита означает, что если бит содержит значение 0, то оно изменяется на 1, а 1 — на 0. Как правило, первоначально IPv6-адреса содержат в этом бите значение 0, инвертируемое в 1.

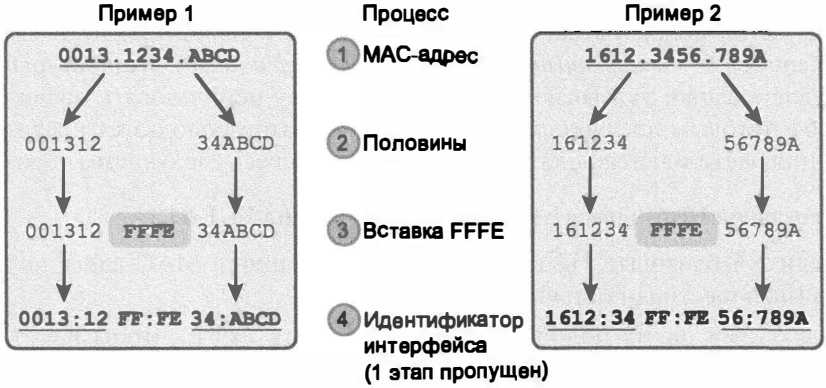


Рис. 27.5. Два примера основной части процесса создания идентификатора интерфейса EUI-64

ВНИМАНИЕ!

Инвертируемый по правилам EUI-64 бит называется *универсальным/локальным битом* (universal/local bit), его значение 0 означает, что MAC-адрес является универсальным встроенным. У всех прошитых MAC-адресов в этой позиции должно быть двоичное значение 0. Поскольку люди редко переопределяют MAC-адреса своих маршрутизаторов, процесс вычисления идентификатора EUI-64, как правило, изменяет седьмой бит с двоичного 0 на двоичную 1.

Пример на рис. 27.6 завершает два примера на рис. 27.5, которые были сосредоточены только на первых двух шестнадцатеричных цифрах. Примеры демонстрируют каждую пару шестнадцатеричных цифр (этап 1) и двоичный эквивалент (этап 2). На этапе 3 представлены копии тех же 8 битов, кроме седьмого инвертируемого бита; пример слева инвертирует 0 в 1, а пример справа инвертирует 1 в 0. И наконец, биты преобразуются назад в шестнадцатеричный формат (этап 4).

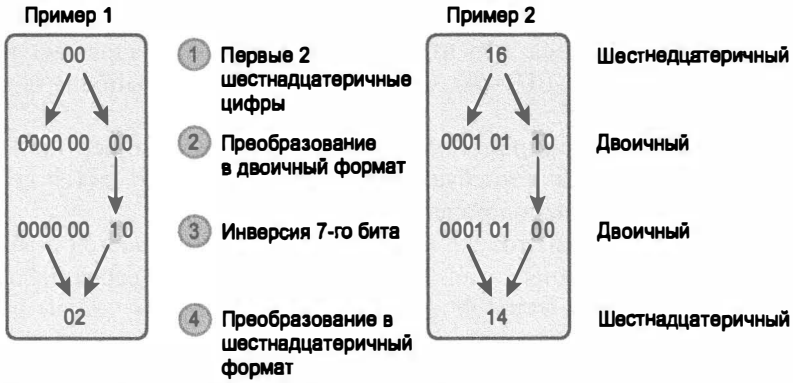


Рис. 27.6. Инверсия седьмого бита поля идентификатора интерфейса EUI-64

ВНИМАНИЕ!

Если вы не помните, как осуществляются преобразования шестнадцатеричных значений в двоичные, уделите время изучению процесса. Если вы запоминаете 16 шестнадцатеричных значений для цифр от 0 до F с соответствующими им двоичными значениями, то преобразование будет очень простым. Если же вы не полагаетесь на свою память, обратитесь к табл. A.2 в приложении A.

Как обычно, наилучший способ научиться создавать идентификаторы интерфейса EUI-64 заключаются в вычислении их самостоятельно. Несколько практических задач с 64-битовым префиксом IPv6 в первом столбце и MAC-адресом во втором столбце приведено в табл. 27.1. Следует вычислить полный (несокращенный) IPv6-адрес, используя правила EUI-64. Ответы находятся в конце главы, в разделе "Ответы на практические задания".

Таблица 27.1. Практические задания на создание IPv6-адресов EUI-64

Префикс	MAC-адрес	Несокращенный IPv6-адрес
2001:DB8:1:1::/64	0013.ABAB.1001	
2001:DB8:1:1::/64	AA13.ABAB.1001	
2001:DB8:1:1::/64	000C.BEEF.CAFE	
2001:DB8:1:1::/64	B80C.BEEF.CAFE	
2001:DB8:FE:FE::/64	0C0C.ABAC.CABA	
2001:DB8:FE:FE::/64	0A0C.ABAC.CABA	

Настройка интерфейса маршрутизатора на использование формата EUI-64 подразумевает использование подкоманды интерфейса `ipv6 address` *адрес/длина_префикса* `eui-64`. Ключевое слово `eui-64` указывает маршрутизатору искать MAC-адрес интерфейса и осуществлять преобразования EUI-64 для вычисления идентификатора интерфейса.

Пересмотренная по сравнению с прежним примером 27.1 конфигурация на маршрутизаторе R1 представлена в примере 27.5. В данном случае, маршрутизатор R1 использует для своих IPv6-адресов формат EUI-64.

Пример 27.5. Настройка IPv6-адресов на интерфейсе маршрутизатора R1 с использованием формата EUI-64

```
! Следующие команды на маршрутизаторе R1
ipv6 unicast-routing
!
! Теперь команда ipv6 address выводит префикс
interface GigabitEthernet0/0
  ipv6 address 2001:DB8:1111:1::/64 eui-64
!
interface Serial0/0/0
  ipv6 address 2001:DB8:1111:2::/64 eui-64

R1# show ipv6 interface brief
GigabitEthernet0/0                    [up/up]
    FE80::1FF:FE01:101
    2001:DB8:1111:1:0:1FF:FE01:101
```

```
GigabitEthernet0/1                      [administratively down/down]
    unassigned
Serial0/0/0                             [up/up]
    FE80::1FF:FE01:101
    2001:DB8:1111:2:0:1FF:FE01:101
Serial0/0/1                             [administratively down/down]
    unassigned
```

Обратите внимание, что пример демонстрирует применение формата EUI-64 на последовательном интерфейсе, с которым не связан никакой MAC-адрес. Для интерфейсов без MAC-адреса маршрутизатор сам выбирает номер MAC самого младшего интерфейса маршрутизатора, обладающего таковым. В этом примере для формирования идентификатора EUI-64 всех последовательных интерфейсов маршрутизатор R1 использует MAC-адрес своего интерфейса G0/0.

ВНИМАНИЕ!

При использовании формата EUI-64 значение адреса в команде `ipv6 address` должно быть префиксом, а не полным 128-битовым IPv6-адресом. Но если по ошибке ввести полный адрес и использовать ключевое слово `eui-64`, то операционная система IOS примет команду и преобразует адрес в префикс прежде, чем поместит команду в выполняющийся файл конфигурации. Например, команда `ipv6 address 2000:1:1:1::1/64 eui-64` преобразуется в команду `ipv6 address 2000:1:1:1::/64 eui-64`.

Динамическая настройка одноадресатного адреса

Как правило, сетевые инженеры настраивают IPv6-адреса интерфейсов маршрутизаторов так, чтобы они не изменились, пока инженер сам не изменит конфигурацию маршрутизатора. Но маршрутизаторы могут быть настроены и на динамическое изучение IPv6-адресов. Это может быть полезно для маршрутизаторов, подключенных к Интернету по модемам DSL и кабельным модемам.

Маршрутизаторы Cisco предоставляют два способа динамического изучения IPv6-адресов для использования интерфейсами маршрутизатора.

- Протокол DHCP с фиксацией состояния.
- Автоматическая настройка адреса без фиксации состояния (SLAAC — Stateless Address Auto Configuration).

Оба способа используют знакомую команду `ipv6 address`. Конечно, фактический IPv6-адрес они не задают, они определяют ключевое слово, указывающее маршрутизатору используемый способ изучения IPv6-адреса. В примере 27.6 приведена конфигурация с одним интерфейсом, использующим протокол DHCP с фиксацией состояния, и другим, использующим протокол SLAAC.

Пример 27.6. Настройка маршрутизатора на изучение IPv6-адресов при помощи интерфейсов DHCP и SLAAC

```
! Этот интерфейс использует для изучения своего IPv6-адреса протокол DHCP
interface FastEthernet0/0
ipv6 address dhcp
!
! Этот интерфейс использует для изучения IPv6-адреса протокол SLAAC
```

```
interface FastEthernet0/1
ipv6 address autoconfig
```

Маршрутизаторы Cisco должны быть также готовы выполнять роль от имени других устройств IPv6 в сети при работе с протоколами DHCP и SLAAC. Реализация протокола IPv6 на хостах, а также сопутствующие протоколы и обязанности маршрутизаторов обсуждаются в главе 28.

Специальные адреса, используемые маршрутизаторами

Настройка протокола IPv6 на маршрутизаторе начинается с простых этапов, обсуждаемых в первой части этой главы. После задания глобальной команды конфигурации `ipv6 unicast-routing`, разрешающей маршрутизацию IPv6, добавление одноадресатного IPv6-адреса на интерфейсе заставляет маршрутизатор осуществлять следующее.

Функции IOS, разрешаемые при настройке протокола IPv6 на рабочем интерфейсе

Ключевая тема

- Предоставляет интерфейсу одноадресатный IPv6-адрес.
- Разрешает маршрутизацию пакетов IPv6 на этом интерфейсе.
- Определяет префикс IPv6 (подсеть) для этого интерфейса.
- Указывает маршрутизатору добавить связанный с этим префиксом маршрут IPv6 в таблицу маршрутизации IPv6, когда он находится в состоянии up/up.

ВНИМАНИЕ!

Фактически, если внимательно рассмотреть список снова, то можно заметить, что это те же концепции, что и у протокола IPv4 при настройке IPv4-адреса на интерфейсе маршрутизатора.

Хотя все средства протокола IPv6 в этом списке подобны таковым у протокола IPv4, у первых есть также много дополнительных функций, отсутствующих в протоколе IPv4. Зачастую эти дополнительные функции используют другие IPv6-адреса, большинство из которых являются многоадресатными адресами. В данном разделе этой главы рассматриваются дополнительные IPv6-адреса, используемые на маршрутизаторах, с кратким описанием их применения.

Адреса, локальные в пределах канала связи

Протокол IPv6 использует *адреса, локальные в пределах канала связи* (link-local address), как специальный вид одноадресатного IPv6-адреса. Эти адреса используются не для обычных пакетов IPv6, содержащих прикладные данные, а для вспомогательных протоколов и маршрутизации. В данном разделе сначала рассматривается использование протоколом IPv6 локальных в пределах канала связи адресов, а затем то, как маршрутизаторы создают их.

Концепция адресов, локальных в пределах канала связи

Каждый хост IPv6 (включенные маршрутизаторы) использует дополнительный одноадресатный адрес, называемый локальным в пределах канала связи. Послан-

ные на этот адрес пакеты не оставляют подсеть IPv6, поскольку маршрутизаторы не перенаправляют их.

Протокол IPv6 использует локальные в пределах канала связи адреса для множества других протоколов. Многие протоколы IPv6, требующие передачи сообщений в одной подсети, как правило, используют адреса, локальные в пределах канала связи, а не глобальные одноадресатные или уникальные локальные адреса хоста. Например, протокол обнаружения соседних устройств (Neighbor Discovery Protocol — NDP), взявший на себя функции протокола ARP из стека IPv4, использует адреса, локальные в пределах канала связи.

Маршрутизаторы IPv6 также используют адреса, локальные в пределах канала связи, как IP-адреса следующих транзитных точек перехода (рис. 27.7). Хосты IPv6 также используют концепцию стандартного маршрутизатора (стандартного шлюза), как и в IPv4, но вместо адреса маршрутизатора в той же подсети они обращаются по адресу, локальному в пределах канала связи маршрутизатора. Команда `show ipv6 route` выводит локальный в пределах канала связи адрес соседнего маршрутизатора, а не глобальный одноадресатный или уникальный локальный одноадресатный адрес.

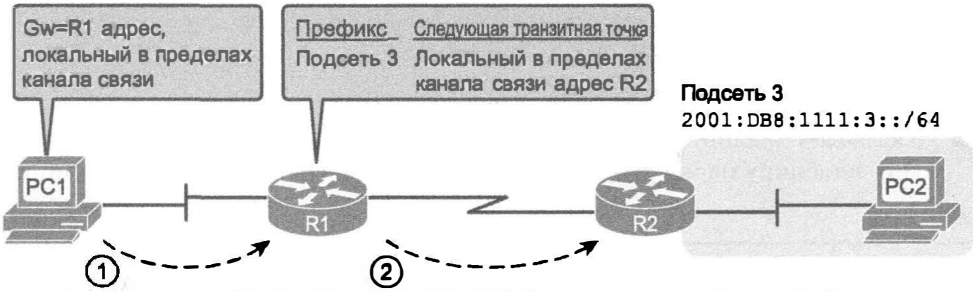


Рис. 27.7. Протокол IPv6 использует адреса, локальные в пределах канала связи, как адреса следующих транзитных точек перехода

Ниже приведены ключевые факты об адресах, локальных в пределах канала связи.

Ключевая тема

Ключевые факты об адресах, локальных в пределах канала связи IPv6

- **Одноадресатный (не многоадресатный)**. Адрес, локальный в пределах канала связи, представляет один хост, и посланные на него пакеты должны быть обработаны только одним хостом IPv6.
- **Область перенаправления локальна только для канала связи**. Пакеты, посланные на локальный в пределах канала связи адрес, не покидают локальный канал связи, поскольку маршрутизаторы не перенаправляют пакеты с локальными в пределах канала связи адресами получателя.
- **Автоматическое создание**. Решая некоторые проблемы инициализации для хостов, прежде чем они динамически изучат глобальный одноадресатный адрес, каждый интерфейс хоста IPv6 (и интерфейс маршрутизатора) может автоматически создать собственный адрес, локальный в пределах канала связи.

- **Общее использование**. Используются некоторыми вспомогательными протоколами, выполняющимися локально в одной подсети, и как адрес следующей транзитной точки перехода маршрутов IPv6.

Создание адресов, локальных в пределах канала связи, на маршрутизаторах

Используя набор простых правил, хосты IPv6 и маршрутизаторы могут вычислить собственный адрес, локальный в пределах канала связи, для каждого интерфейса. Во-первых, все адреса, локальные в пределах канала связи, начинаются с того же префикса, как показано на рис. 27.8, *слева*. По определению первые 10 битов должны соответствовать префиксу FE80::/10, означая, что первыми тремя шестнадцатеричными цифрами будут FE8, FE9, FEA или FEB. Однако, согласно документу RFC, следующие 54 бита должны быть двоичными нулями, поэтому локальный в пределах канала связи адрес должен всегда начинаться с первых четырех несокращенных квартетов FE80:0000:0000:0000.

Рис. 27.8. Формат адреса, локального в пределах канала связи

Вторая половина локального в пределах канала связи адреса фактически может быть сформирована по другим правилам. Для создания идентификатора интерфейса маршрутизаторы Cisco используют формат EUI-64 (см. выше). В результате полный локальный в пределах канала связи адрес маршрутизатора окажется уникальным, поскольку участвующий в процессе EUI-64 MAC-адрес является уникальным. Другие операционные системы создают идентификатор интерфейса случайным образом. Например, операционная система OS Microsoft использует вероятностный процесс для выбора идентификатора интерфейса и изменяет его время от времени, затрудняя попытки атак некоторых форм. И наконец, локальные в пределах канала связи адреса могут быть просто заданы.

Операционная система IOS создает локальный в пределах канала связи адрес для любого интерфейса, на котором командой `ipv6 address` настроен по крайней мере один другой одноадресатный адрес (глобальный одноадресатный или уникальный локальный). Чтобы просмотреть локальный в пределах канала связи адрес, достаточно использовать обычные команды, выводящие одноадресатный IPv6-адрес: `show ipv6 interface` и `show ipv6 interface brief`. Адреса для маршрутизатора R1 показаны в примере 27.7.

Пример 27.7. Сравнение адресов, локальных в пределах канала связи, с созданными одноадресатными адресами EUI

```
R1# show ipv6 interface brief
GigabitEthernet0/0              [up/up]
        FE80::1FF:FE01:101
        2001:DB8:1111:1:0:1FF:FE01:101
GigabitEthernet0/1              [administratively down/down]
        unassigned
```

```
Serial0/0/0                          [up/up]
        FE80::1FF:FE01:101
        2001:DB8:1111:2:0:1FF:FE01:101
Serial0/0/1                          [administratively down/down]
        unassigned
```

Сначала рассмотрим две пары элементов, выделенных в примере. Для каждого из двух интерфейсов, обладающих глобальным одноадресатным адресом (G0/0 и S0/0/0), вывод указывает глобальный одноадресатный адрес, в данном случае начинающийся на 2001. В то же время вывод указывает также для каждого интерфейса адрес, локальный в пределах канала связи (начинающийся на FE80).

Теперь сосредоточимся на двух адресах, выведенных для интерфейса G0/0. Если внимательно посмотреть на вторую половину двух представленных для интерфейса G0/0 адресов, то можно заметить, что у обоих адресов одинаковое значение идентификатора интерфейса. Глобальный одноадресатный адрес был задан в данном случае командой `ipv6 address 2001:DB8:1111:1::/64 eui-64`, поэтому для формирования глобального одноадресатного и локального в пределах канала связи адреса маршрутизатор использовал логику EUI-64, в данном случае MAC-адрес интерфейса 0200.0101.0101. Поэтому маршрутизатор вычисляет часть идентификатора интерфейса обоих адресов как 0000:01FF:FE01:0101 (несокращенный). После сокращения локальный в пределах канала связи адрес маршрутизатора R1 на интерфейсе G0/0 становится FE80::1FF:FE01:101.

Операционная система IOS может создать локальный в пределах канала связи адрес автоматически, но он может быть и задан. Операционная система IOS выбирает для интерфейса локальный в пределах канала связи адрес на основании следующих правил.

- Если задано, маршрутизатор использует значение из подкоманды интерфейса `ipv6 address` *адрес* `link-local`. Обратите внимание, что заданный локальный в пределах канала связи адрес должен относиться к корректному диапазону локальных в пределах канала связи адресов, т.е. адресов префикса FE80::/10. Другими словами, адрес должен начинаться с FE8, FE9, FEA или FEB.
- Если не задано, операционная система IOS вычислит локальный в пределах канала связи адрес, используя правила EUI-64 (см. пример 27.7). Вычисление использует правила EUI-64, даже если одноадресатный адрес интерфейса не использует формат EUI-64.

Многоадресатные IPv6-адреса

Протокол IPv6 использует многоадресатные IPv6-адреса в нескольких целях. В некоторых случаях вспомогательные протоколы используют эти адреса для передачи пакетов на несколько хостов IPv6 сразу. В других случаях приложения используют многоадресатные адреса для передачи одного пакета IPv6 на многие хосты, а не отдельного пакета на каждый хост индивидуально.

В следующем разделе сначала сравнивается применение широковещательного адреса с применением многоадресатных адресов. Затем рассматриваются некоторые общепринятые многоадресатные адреса, используемые протоколом IPv6.

Широковещательные адреса или многоадресатные

У терминов *широковещательный* (broadcast) и *многоадресатный* (multicast) есть хоть и небольшое, но различие. После более подробного изучения обоих, особенно в условиях групповой передачи, различия станут очевидней. Но поскольку экзамены CCENT и CCNA R/S не уделяют большого внимания этой теме, рассмотрим ее кратко.

Сначала рассмотрим протокол IPv4 и подсеть IPv4. Протокол IPv4 использует широковещание, т.е. протокол IPv4 позволяет хосту посылать пакеты на широковещательный адрес, поэтому все хосты IPv4 в этой подсети должны прослушивать эти пакеты и читать их, тратя несколько циклов процессора на принятие решения о том, следует ли ответить. Короче говоря, широковещательный пакет поступает на все устройства, и все они должны его рассмотреть.

Широковещание может быть весьма расточительным по циклам процессора на хостах в подсети, поскольку зачастую оно используется для получения ответа только от одного или нескольких хостов в этой подсети. Групповая передача помогает решить эту проблему, позволяя обрабатывать пакет только заданным подмножеством хостов. Поскольку групповая передача эффективней для устройств в сети, протокол IPv6 использует вместо широковещания многоадресатные адреса.

Для завершенности сравнения обдумаем теперь протокол IPv6 и подсеть IPv6 со 100 хостами и 3 маршрутизаторами. Иногда хост должен послать пакет на "все хосты, поддерживающие протокол IPv6". Для этого протокол IPv6 определяет многоадресатный адрес (FF02::1). Посланные на адрес FF02::1 пакеты поступают на все устройства, поддерживающие протокол IPv6 (включая маршрутизаторы), и должны обрабатываться ими всеми. Но если сообщение протокола должно попасть на все маршрутизаторы IPv6, но не на хосты, то пакет можно послать на адрес FF02::2 — многоадресатный адрес, используемый только маршрутизаторами IPv6. Посланные на этот адрес пакеты не требуют обработки от устройств, не являющихся маршрутизаторами, экономя циклы их процессоров.

Кроме того, пакет можно послать на некое подмножество маршрутизаторов. Расширенный протокол маршрутизации внутреннего шлюза (Enhanced Interior Gateway Routing Protocol — EIGRP) использует зарезервированный многоадресатный адрес (FF02::A), а открытый протокол поиска первого кратчайшего маршрута (Open Shortest Path First — OSPF) использует два адреса (FF02::5 и FF02::6). Эти протоколы маршрутизации посылают свои сообщения об обновлении на эти зарезервированные многоадресатные адреса, поэтому маршрутизатор, выполняющий протокол EIGRP, не беспокоит маршрутизатор, выполняющий протокол OSPF, и наоборот.

Общепринятые многоадресатные адреса локальной области видимости

В одних случаях многоадресатные сообщения должны оставаться в пределах одной подсети, а в других случаях они должны распространяться на многие подсети. Некоторые многоадресатные адреса подразумевают, что посланный на них пакет должен остаться в пределах канала связи; у этих адресов локальная область видимости канала связи. У многоадресатных адресов, позволяющих перенаправлять пакеты в другие подсети предприятия, локальная область видимости организации.

В этой книге представлено несколько общепринятых адресов с локальной областью видимости канала связи. Агентство (Internet Assigned Numbers Authority — IANA) резервирует для групповой передачи все IPv6-адреса, начинающиеся на FF, или, бо-

лее формально, имеющие префикс FF00::/8. В пределах того диапазона агентство IANA резервирует все начинающиеся на FF02 адреса (формально FF02::/16) для многоадресатных адресов локальной области видимости канала связи.

В табл. 27.2 приведены наиболее распространенные многоадресатные IPv6-адреса с локальной областью видимости.

Таблица 27.2. Ключевые многоадресатные IPv6-адреса с локальной областью видимости

Короткое название	Много-адресатный адрес	Значение	Эквивалент IPv4
Все узлы (all-nodes)	FF02::1	Все интерфейсы на канале связи, использующие протокол IPv6	Широковещательный адрес подсети
Все маршрутизаторы (all-routers)	FF02::2	Все интерфейсы маршрутизатора IPv6 на канале связи	Отсутствует
Все OSPF Все OSPF-DR	FF02::5, FF02::6	Все маршрутизаторы OSPF и все выделенные маршрутизаторы OSPF соответственно	224.0.0.5, 224.0.0.6
Маршрутизаторы EIGRPv6	FF02::A	Все маршрутизаторы, использующие протокол EIGRP для протокола IPv6 (EIGRPv6)	224.0.0.10

Пример 27.8 повторяет вывод команды `show ipv6 interface`, чтобы показать многоадресатные адреса, используемые маршрутизатором R1 на его интерфейсе G0/0. В данном случае выделенные строки представляют адреса всех узлов (FF02::1), всех маршрутизаторов (FF02::2) и EIGRPv6 (FF02::A).

Пример 27.8. Проверка статических IPv6-адресов на маршрутизаторе R1

```
R1# show ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1FF:FE01:101
  No Virtual link-local address(es):
  Description: LAN at Site 1
  Global unicast address(es):
    2001:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::A
    FF02::1:FF00:1
    FF02::1:FF01:101
! Строки пропущены для краткости
```

Многоадресатные адреса опрашиваемого узла

Большинство многоадресатных адресов, используемых в соответствии с протоколами в пределах экзаменов CCENT и CCNA, является просто числами, зарезервированными документами RFC. Следует просто запомнить эти числа и обращать на них внимание в командах `show`. Однако один специфический тип многоадресатного адреса, *многоадресатный адрес опрашиваемого узла* (solicited-node multicast ad-

dress), отличается от хоста к хосту, поэтому его значение не задано предварительно. В данном разделе кратко рассматривается этот тип многоадресатного адреса.

У некоторых многоадресатных IPv6-адресов есть четкие, осмысленные и короткие названия. Такие термины, как *все узлы* (для многоадресатного адреса FF02::1) или *все маршрутизаторы* (для многоадресатного адреса FF02::2), в значительной степени определяют их значение. Откровенно говоря, термин *опрашиваемый узел* (solicited-node) не отражает идею этого многоадресатного адреса.

По правде говоря, многоадресатный адрес опрашиваемого узла представляет адрес получателя так, чтобы один пакет можно было послать в одной подсети IPv6 (локальной для канала связи) на все хосты, у одноадресатных адресов которых то же значение в последних шести шестнадцатеричных цифрах. Первое предложение, конечно, относительно длинно. Список ниже содержит концепции, фактически определяющие многоадресатный адрес опрашиваемого узла для определенного хоста.

- Многоадресатный адрес (а не одноадресатный).
- Локальная область видимости канала связи.
- Основан на одноадресатном IPv6-адресе хоста.
- Основан именно на последних шести шестнадцатеричных цифрах одноадресатного адреса.
- Каждый хост должен прослушивать пакеты, посланные на его многоадресатный адрес опрашиваемого узла.
- Все хосты, одноадресатные IPv6-адреса которых имеют то же значение в последних шести шестнадцатеричных цифрах, используют тот же многоадресатный адрес опрашиваемого узла и обрабатывают те же пакеты.

Последний элемент списка относится к ключевой функции многоадресатных адресов опрашиваемого узла. Посланные на специфический многоадресатный адрес опрашиваемого узла пакеты могут поступать только на один хост. Но если у нескольких хостов в подсети одинаковые значения в последних шести шестнадцатеричных цифрах одноадресатных адресов, то тот же многоадресатный пакет обрабатывается обоими (или всеми) хостами. Такая логика передачи одного многоадресатного пакета на все хосты с подобными одноадресатными IPv6-адресами используется некоторыми протоколами. Так появился многоадресатный адрес опрашиваемого узла.

ВНИМАНИЕ!

Альтернативное название, такое как многоадресатный, локальный для канала связи адрес, общий для всех хостов с одинаковыми последними шестью шестнадцатеричными цифрами их одноадресатных адресов, было бы более понятным, чем адрес опрашиваемого узла. Но поскольку его значение теперь известно, осталось только запомнить истинное название.

Все хосты IPv6 должны прослушивать сообщения, посланные на их многоадресатный адрес (адреса) опрашиваемого узла. Таким образом, для каждого интерфейса и каждого одноадресатного адреса на каждом интерфейсе устройство должно установить его многоадресатный адрес (адреса) опрашиваемого узла и прослушивать пакеты, посланные на эти адреса.

Логика поиска многоадресатного адреса опрашиваемого узла довольно проста, когда уже известен одноадресатный адрес. Начнем с предопределенного префикса /104, представленного на рис. 27.9. Другими словами, все многоадресатные адреса опрашиваемого узла начинаются с сокращенного FF02::1:FF. В последние 24 бита адреса опрашиваемого узла (6 шестнадцатеричных цифр) скопируйте одноадресатный адрес.

Рис. 27.9. Формат многоадресатного адреса опрашиваемого узла

Чтобы просмотреть примеры этих адресов на маршрутизаторе, вернитесь к примеру 27.8. Последние две строки вывода команды представляют многоадресатные адреса опрашиваемого узла для интерфейса G0/0 маршрутизатора R1: FF02::1:FF00:1 и FF02::1:FF01:101. Обратите внимание, что в данном случае по некоторым причинам у интерфейса G0/0 маршрутизатора R1 есть два таких адреса, и каждый из них соответствует глобальному одноадресатному адресу маршрутизатора на этом интерфейсе, в то время как другие соответствуют адресам, локальным в пределах канала связи (одноадресатным).

Другие IPv6-адреса

Совместно в этой и предыдущей главе представлено большинство концепций IPv6-адресов, рассматриваемых в данной книге. Этот короткий раздел рассматривает некоторые их оставшихся IPv6-адресов и резюмирует их концепции.

Все хосты IPv6 могут использовать два дополнительных специальных адреса.

Ключевая тема **Другие специальные IPv6-адреса**

- Неизвестный (неопределенный) IPv6-адрес :: или все нули.
- Петлевой IPv6-адрес ::1 или 127 двоичных нулей с одной единицей.

Неизвестный адрес (::) может быть использован хостом, когда его собственный IPv6-адрес еще не известен или когда хост подозревает о наличии проблем с его собственным IPv6-адресом. Например, на ранних рабочих этапах динамического обнаружения IPv6-адреса хост использует неизвестный адрес. Когда хост еще не знает, какой IPv6-адрес использовать, он может использовать адрес :: как свой первоначальный IPv6-адрес.

Локальный диагностический IPv6-адрес предоставляет каждому хосту IPv6 способ проверки его собственного стека протоколов. Точно так же как и локальный диагностический адрес IPv4 127.0.0.1, пакеты, посланные на адрес ::1, не покидают хост, а просто доставляются вниз и вверх по стеку IPv6 приложения на локальном хосте.

Обзор

Резюме

- Маршрутизаторы Cisco предоставляют две возможности для статической настройки IPv6-адресов. В одном случае задается полный 128-битовый адрес, в другом задается лишь 64-битовый префикс, а вторую половину адреса (идентификатор интерфейса) маршрутизатор получает сам.
- Для статического задания полного 128-битового одноадресатного адреса (или глобального одноадресатного, или уникального локального адреса) маршрутизатор нуждается в подкоманде интерфейса `ipv6 address` *адрес/длина_префикса* на каждом интерфейсе.
- Обратите внимание, что маршрутизация IPv6 должна быть разрешена на маршрутизаторе как глобально (`ipv6 unicast-routing`), так и на интерфейсе (`ipv6 address`), прежде чем он попытается перенаправить пакеты через этот интерфейс. (Если на маршрутизаторе случайно пропущена команда `ipv6 unicast-routing`, его интерфейсы могут быть полностью настроены IPv6-адресами, но маршрутизатор действует как хост IPv6 и не перенаправляет пакеты IPv6.)
- Протокол IPv6 использует множество команд `show`, подражающих синтаксису команд `show` протокола IPv4. Соответствующие примеры приведены ниже.
  - Команда `show ipv6 interface brief` предоставляет информацию об IPv6-адресе интерфейса, но не о длине префикса, как подобная команда `show ip interface brief` протокола IPv4.
  - Команда `show ipv6 interface` предоставляет подробности о параметрах IPv6 интерфейса, что очень похоже на команду `show ip interface` протокола IPv4.
- Для создания уникального идентификатора интерфейса с использованием правил EUI-64 необходимо следующее.
  - Настройка интерфейса маршрутизатора на использование формата EUI-64 подразумевает использование подкоманды интерфейса `ipv6 address` *адрес/длина_префикса* `eui-64`. Ключевое слово `eui-64` указывает маршрутизатору искать MAC-адрес интерфейса и осуществлять преобразования EUI-64 для вычисления идентификатора интерфейса.
  - Далее маршрутизатор использует правила EUI-64 для создания части идентификатора интерфейса в адресе следующим образом.
  - Разделите 6-байтовый (12 шестнадцатеричных цифр) MAC-адрес на две половины (по 6 шестнадцатеричных цифр каждая).
  - Между этими двумя половинами вставьте часть FFFE, чтобы идентификатор интерфейса имел в общей сложности 16 шестнадцатеричных цифр (64 бита).
  - Инвертируйте седьмой бит идентификатора интерфейса.

- Маршрутизаторы Cisco предоставляют два способа динамического изучения IPv6-адресов для использования интерфейсами маршрутизатора.
  - Протокол DHCP с фиксацией состояния.
  - Автоматическая настройка адреса (SLAAC) без фиксации состояния.
- После задания глобальной команды конфигурации `ipv6 unicast-routing`, разрешающей маршрутизацию IPv6, добавление одноадресатного IPv6-адреса на интерфейсе заставляет маршрутизатор осуществлять следующее.
  - Предоставляет интерфейсу одноадресатный IPv6-адрес.
  - Разрешает маршрутизацию пакетов IPv6 на этом интерфейсе.
  - Определяет префикс IPv6 (подсеть) для этого интерфейса.
  - Указывает маршрутизатору добавить связанный с этим префиксом маршрут IPv6 в таблицу маршрутизации IPv6, когда он находится в состоянии up/up.
- Каждый хост IPv6 (включенные маршрутизаторы) использует дополнительный одноадресатный адрес, называемый локальным в пределах канала связи. Посланные на этот адрес пакеты не оставляют подсеть IPv6, поскольку маршрутизаторы не перенаправляют их.
  - **Одноадресатный (не многоадресатный)**. Адрес, локальный в пределах канала связи, представляет один хост, и посланные на него пакеты должны быть обработаны только одним хостом IPv6.
  - **Область перенаправления локальна только для канала связи**. Пакеты, посланные на локальный в пределах канала связи адрес, не покидают локальный канал связи, поскольку маршрутизаторы не перенаправляют пакеты с локальными в пределах канала связи адресами получателя.
  - **Автоматическое создание**. Решая некоторые проблемы инициализации для хостов, прежде чем они динамически изучат глобальный одноадресатный адрес, каждый интерфейс хоста IPv6 (и интерфейс маршрутизатора) может автоматически создать собственный адрес, локальный в пределах канала связи.
  - **Общее использование**. Используются некоторыми вспомогательными протоколами, выполняющимися локально в одной подсети, и как адрес следующей транзитной точки перехода маршрутов IPv6.
- Все хосты IPv6 могут использовать два дополнительных специальных адреса.
  - Неизвестный (неопределенный) IPv6-адрес :: или все нули.
  - Петлевой IPv6-адрес ::1 или 127 двоичных нулей с одной единицей.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. У маршрутизатора R1 есть интерфейс Gigabit Ethernet 0/1 с установленным MAC-адресом 0200.0001.000A. Какая из следующих команд, введенных в режи-

ме конфигурации интерфейса, присвоит интерфейсу G0/1 этого маршрутизатора одноадресатный IPv6-адрес 2001:1:1:1:1:200:1:A с длиной префикса /64?

А) `ipv6 address 2001:1:1:1:1:200:1:A/64`

Б) `ipv6 address 2001:1:1:1:1:200:1:A/64 eui-64`

В) `ipv6 address 2001:1:1:1:1:200:1:A /64 eui-64`

Г) `ipv6 address 2001:1:1:1:1:200:1:A /64`

Д) Все ответы не правильные.

2. У маршрутизатора R1 есть интерфейс Gigabit Ethernet 0/1 с установленным MAC-адресом 5055.4444.3333. Этот интерфейс был настроен подкомандой `ipv6 address 2000:1:1:1::/64 eui-64`. Какой одноадресатный адрес использует этот интерфейс?

А) 2000:1:1:1:52FF:FE55:4444:3333

Б) 2000:1:1:1:5255:44FF:FE44:3333

В) 2000:1:1:1:5255:4444:33FF:FE33

Г) 2000:1:1:1:200:FF:FE00:0

3. Маршрутизатор R1 в настоящее время поддерживает протокол IPv4, перенаправляя пакеты через все свои интерфейсы. Конфигурация маршрутизатора R1 должна быть переведена на поддержку двойного стека, обеспечивающего и маршрутизацию IPv4, и IPv6. Какая из следующих задач должна быть выполнена прежде, чем маршрутизатор сможет поддерживать также маршрутизацию пакетов IPv6? (Выберите два ответа.)

А) Разрешить протокол IPv6 на каждом интерфейсе, используя подкоманду интерфейса `ipv6 address`.

Б) Разрешить поддержку обеих версий при помощи глобальной команды `ip versions 4 6`.

В) Дополнительно разрешить маршрутизацию IPv6, используя глобальную команду `ipv6 unicast-routing`.

Г) Перейти на двойной стек маршрутизации, используя глобальную команду `ip routing dual-stack`.

4. У маршрутизатора R1 есть интерфейс Gigabit Ethernet 0/1 с установленным MAC-адресом 0200.0001.000A. Далее интерфейс настроен подкомандой интерфейса `ipv6 address 2001:1:1:1:200:FF:FE01:B/64`; никаких других команд `ipv6 address` на интерфейсе не введено. В каком из следующих ответов приведен локальный в пределах канала связи адрес, используемый на интерфейсе?

А) `FE80::FF:FE01:A`

Б) `FE80::FF:FE01:B`

В) `FE80::200:FF:FE01:A`

Г) `FE80::200:FF:FE01:B`

5. Какой из следующих многоадресатных адресов определяется как адрес для передачи пакетов только маршрутизаторам IPv6 на локальном канале связи?

А) FF02::1

Б) FF02::2

В) FF02::5

Г) FF02::A

6. У маршрутизатора R1 есть интерфейс Gigabit Ethernet 0/1 с установленным MAC-адресом 0200.0001.000A. Какая из следующих команд, введенных в режиме конфигурации интерфейса, присвоит интерфейсу G0/1 этого маршрутизатора многоадресатный адрес опрашиваемого узла FF02::1:FF00:A?

А) `ipv6 address 2001:1:1:1:1:200:1:A/64`

Б) `ipv6 address 2001:1:1:1::/64 eui-64`

В) `ipv6 address 2001:1:1:1::A/64`

Г) Ни одна из команд не присвоит этот многоадресатный адрес.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 27.3.

Таблица 27.3. Ключевые темы главы 27

Элемент	Описание	Страница
Рис. 27.2	Вероятный способ использования двух стеков (IPv4 и IPv6) на протяжении переходного периода	776
Список	Правила создания IPv6-адреса с использованием правил EUI-64	781
Рис. 27.4	Формат IPv6-адреса с идентификатором интерфейса и частью EUI-64	781
Рис. 27.6	Инверсия седьмого бита поля идентификатора интерфейса EUI-64	782
Список	Функции IOS, разрешаемые при настройке протокола IPv6 на рабочем интерфейсе	785
Список	Ключевые факты об адресах, локальных в пределах канала связи IPv6	786
Рис. 27.9	Формат многоадресатного адреса опрашиваемого узла	792
Список	Другие специальные IPv6-адреса	792

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

двойной стек (dual stack), EUI-64, адрес, локальный в пределах канала связи (link-local address), локальная область видимости канала связи (link-local scope), многоадресатный адрес опрашиваемого узла (solicited-node multicast address), многоадресатный адрес всех узлов (all-nodes multicast address), многоадресатный адрес всех маршрутизаторов (all-routers multicast address)

Практика

Дополнительная практика на сокращения IPv6-адресов

Для дополнительной практики по поиску адресов EUI-64 и многоадресатных адресов опрашиваемого узла.

- Приложение M (Appendix L) на веб-сайте содержит набор дополнительных практических заданий на вычисление адресов EUI-64 и многоадресатных адресов опрашиваемого узла.
- Создайте собственные задания, используя любой реальный маршрутизатор или эмулятор. Включите интерфейс CLI маршрутизатора в режиме конфигурации и введите команды `mac-address` *адрес* и `ipv6 address` *префикс*`/64 eui-64`. Эти команды (соответственно) изменяют используемый на интерфейсе MAC-адрес и указывают маршрутизатору создавать IPv6-адрес, используя правила EUI-64. Затем, прежде чем подсмотреть IPv6-адрес, выбранный маршрутизатором, осуществите вычисления самостоятельно. И наконец, используйте команду `show ipv6 interface` для просмотра одноадресатного адреса, а также многоадресатного адреса опрашиваемого узла.

Таблицы команд

Хотя и не обязательно заучивать информацию из таблиц данного раздела, в нем приведен список команд конфигурации и пользовательских команд главы. Фактически команды стоит запомнить, чтобы лучше понять материал главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 27.4. Команды конфигурации главы 27

Команда	Описание
`ipv6 unicast-routing`	Глобальная команда, разрешающая маршрутизацию IPv6 на маршрутизаторе
`ipv6 address` *ipv6-адрес/длина_префикса* `[eui-64]`	Подкоманда интерфейса, позволяющая вручную задать весь IP-адрес интерфейса или префикс /64 маршрутизатору, автоматически создающему идентификатор интерфейса в формате EUI-64

Таблица 27.5. Пользовательские команды главы 27

Команда	Описание
`show ipv6 route [connected]`	Выводит маршруты IPv6 или только подключенные маршруты
`show ipv6 interface [`*тип номер*`]`	Выводит параметры IPv6 на интерфейсе, включая локальные для канала связи и другие одноадресатные IP-адреса
`show ipv6 interface brief`	Выводит состояние и IPv6-адреса для каждого интерфейса

Ответы на практические задания

В табл. 27.6 приведены ответы на приведенные ранее в этой главе практические задания по вычислению IPv6-адреса на основании правил EUI-64.

Таблица 27.6. Ответы на практические задания по созданию IPv6-адресов EUI-64

Префикс	MAC-адрес	Несокращенный IPv6-адрес
2001:DB8:1:1::/64	0013.ABAB.1001	2001:DB8:1:1:0213:ABFF:FEAB:1001
2001:DB8:1:1::/64	AA13.ABAB.1001	2001:DB8:1:1:A813:ABFF:FEAB:1001
2001:DB8:1:1::/64	000C.BEEF.CAFE	2001:DB8:1:1:020C:BEFF:FEEF:CAFE
2001:DB8:1:1::/64	B80C.BEEF.CAFE	2001:DB8:1:1:BA0C:BEFF:FEEF:CAFE
2001:DB8:FE:FE::/64	0C0C.ABAC.CABA	2001:DB8:FE:FE:0E0C:ABFF:FEAC:CABA
2001:DB8:FE:FE::/64	0A0C.ABAC.CABA	2001:DB8:FE:FE:080C:ABFF:FEAC:CABA

Ответы на контрольные вопросы:

1 А. **2** Б. **3** А и В. **4** А. **5** Б. **6** В.

ГЛАВА 28

Реализация IPv6-адресации на хостах

Работа хостов IPv6 во многом совпадает с работой хостов IPv4 — используются подобные идеи, подобные протоколы и даже подобные или идентичные команды для тех же целей. В то же время протокол IPv6 иногда использует не такие подходы, как протокол IPv4, и множество иных решений, обусловленных новыми протоколами или командами. Например:

- Подобно хостам IPv4, хосты IPv6 используют одноадресатный адрес, длину префикса (маску), адрес стандартного маршрутизатора и сервера DNS.
- Подобно хостам IPv4, хосты IPv6 используют протокол для динамического изучения MAC-адресов других хостов в той же подсети на базе LAN.
- В отличие от хостов IPv4, хосты IPv6 используют протокол обнаружения соседних устройств (Neighbor Discovery Protocol — NDP) для многих функций, включая функции протокола ARP у IPv4.
- Подобно хостам IPv4, хосты IPv6 могут использовать протокол DHCP для изучения четырех своих главных параметров IPv6.
- В отличие от протокола IPv4, протокол IPv6 поддерживает процесс динамического присвоения адресов, отличный от DHCP, под названием *автоматическая настройка адреса* (Stateless Address Autoconfiguration — SLAAC).

Эта глава посвящена четырем главным параметрам IPv6 хоста: адресу, длине префикса, адресу стандартного маршрутизатора и адресу сервера DNS. Но чтобы понять, как хосты динамически изучают эти адреса, в первом разделе описан протокол NDP, играющий ключевую роль в некоторых процессах IPv6. В следующем разделе речь пойдет о том, как хосты динамически изучают свои параметры IPv6 при помощи протоколов DHCP и SLAAC. В заключительном разделе рассматриваются инструментальные средства проверки большинства параметров хоста IPv6, использующих те же команды, что и в протоколе IPv4.

В этой главе рассматриваются следующие экзаменационные темы

Технологии коммутации сетей LAN

Проверка состояния сети и работоспособности коммутатора с помощью базовых сетевых утилит ping, telnet и ssh.

IP-адресация (IPv4/IPv6)

Технологические требования для запуска протокола IPv6 совместно с протоколом IPv4 как двойного стека.

Описание IPv6-адреса.

Автоматическая настройка.

Технологии маршрутизации IP

Проверка конфигурации маршрутизатора и сетевого подключения.

Службы IP

Настройка и проверка DHCP (маршрутизатор IOS).

Настройка интерфейса маршрутизатора для использования DHCP.

Поиск и устранение неисправностей

Поиск и устранение распространенных проблем, связанных с настройкой хоста и IP-адресации.

Основные темы

Протокол обнаружения соседних устройств

Хостам IPv6 должно быть известно несколько важных параметров IPv6, аналогичных таковым у хостов IPv4: адрес, длина префикса (эквивалент маски), адрес стандартного маршрутизатора и адрес (адреса) сервера DNS. Все четыре концепции, используемые на компьютере PC1, приведены на рис. 28.1, *слева*.

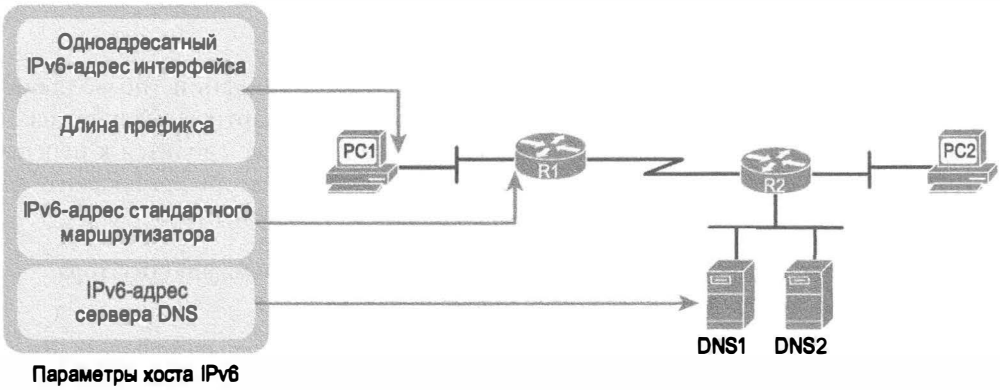


Рис. 28.1. Необходимые хостам параметры IPv6

Обратите внимание: тремя из этих четырех параметров являются одноадресатные IPv6-адреса. Собственный IPv6-адрес компьютера — это, как правило, глобальный одноадресатный или уникальный локальный одноадресатный адрес, используемый компьютером при обращении к серверам DNS. Но поскольку стандартный маршрутизатор должен быть доступен локально, ему обычно присваивается адрес, локальный в пределах канала связи маршрутизатора.

Протокол обнаружения соседних устройств (NDP) определяет несколько разных функций, связанных с адресацией IPv6, следующим образом.

Четыре функции, для которых протокол NDP играет главную роль

Ключевая тема

- **SLAAC**. При использовании автоматической настройки адреса (SLAAC) хост использует сообщения NDP для изучения первой части его адреса и длины префикса.
- **Поиск маршрутизатора**. Используя сообщения NDP, хосты изучают IPv6-адреса доступных маршрутизаторов IPv6 в той же подсети.
- **Обнаружение конфликта адресов**. Независимо от того, как хост устанавливает или изучает свой IPv6-адрес, он не применяет его, пока не убедится, что никакой другой хост не использует тот же адрес. Как хост решает эту проблему? Используя сообщения NDP, конечно, и процесс *обнаружения конфликта адресов* (Duplicate Address Detection — DAD).

- **Обнаружение MAC-адресов соседних хостов**. После прохождения процесса DAD и начала использования своего IPv6-адреса, LAN-ориентированный хост должен изучить MAC-адреса других хостов в той же подсети. Протокол NDP заменяет протокол ARP стека IPv4, поддерживая сообщения, заменяющие сообщения запроса и ответа протокола ARP.

Остальная часть этого раздела посвящена каждой из этих четырех функций. А обсуждение процесса SLAAC отложим до последующей части главы, сосредоточившись в этом разделе больше на базовых функциях NDP.

Обнаружение маршрутизаторов при помощи сообщений NDP RS и RA

Протокол NDP определяет соответствующую пару сообщений, позволяющих хосту динамически обнаруживать все потенциальные стандартные маршрутизаторы, находящиеся на том же канале связи. Упрощенно процесс сводится к передаче хостом многоадресатного сообщения с запросом "маршрутизаторы, расскажите мне о себе" и ответе маршрутизаторов. Возможны следующие сообщения.

Ключевая тема

Описания запросов на получение информации о наличии маршрутизатора NDP и сообщений анонсирования маршрутизатора

- *Запрос на получение информации о наличии маршрутизатора* (Router Solicitation — RS). Сообщение, передаваемое на многоадресатный адрес локальной области видимости FF02::2 ("все маршрутизаторы IPv6"), чтобы опросить все маршрутизаторы только на локальном канале связи и попросить их идентифицировать себя.
- *Анонс маршрутизатора* (Router Advertisement — RA). Сообщение, передаваемое маршрутизатором и содержащее множество фактов, включая локальный для канала связи IPv6-адрес маршрутизатора. Не будучи запрошенным, оно передается на многоадресатный адрес локальной области видимости FF02::1 ("все хосты IPv6"). Будучи послано в ответ на сообщение RS, оно передается либо назад, на одноадресатный адрес пославшего запрос хоста, либо на адрес FF02::1 ("все хосты IPv6").

Например, на рис. 28.2 представлен хост PC1, способный изучить адрес, локальный в пределах канала связи маршрутизатора R1. Процесс действительно прост: компьютер PC1 запрашивает маршрутизатор R1 и получает ответ.

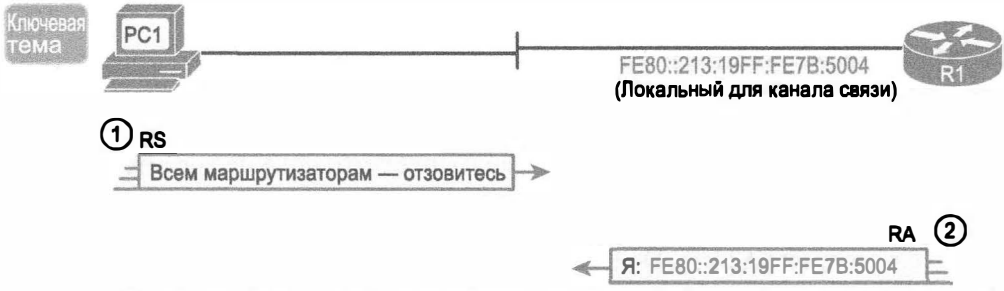


Рис. 28.2. Пример процесса NDP RS/RA по поиску стандартных маршрутизаторов

ВНИМАНИЕ!

Протокол IPv6 допускает указание в сообщении RA нескольких префиксов и нескольких стандартных маршрутизаторов; на рисунке для простоты представлено только по одному из них.

Протокол IPv6 не использует широковещание, но он использует групповую передачу. В данном случае сообщение RS передается на многоадресатный адрес для всех маршрутизаторов (FF02::2), чтобы его получили все маршрутизаторы. Результат будет тот же, что и при широковещании IPv4, но без негативных последствий. В данном случае на обработку сообщений RS циклы процессора потратят только маршрутизаторы IPv6. Сообщение RA может быть передано или на одноадресатный IPv6-адрес компьютера PC1, или на адрес FF02::1 для всех узлов.

Обратите внимание, что хотя на рис. 28.2 представлено, как хост может узнавать о любых маршрутизаторах, маршрутизаторы также сами периодически посылают сообщения RA, даже без входящего сообщения RS. Когда маршрутизаторы посылают эти регулярные сообщения RA, они в основном анонсируют подробности о параметрах IPv6 на канале связи. В данном случае сообщения RA передаются на многоадресатный IPv6-адрес FF02::1, т.е. на все хосты.

Обнаружение информации об адресах для процесса SLAAC при помощи сообщений NDP RS и RA

Хотя сообщения NDP, RS и RA идентифицируют маршрутизаторы IPv6, они поставляют также и другую информацию. Если подумать чуть шире, то сообщения RS позволяют хосту запросить более общую информацию, например: "Маршрутизаторы IPv6, сообщите мне известную вам информацию!" Сообщение RA позволяет маршрутизаторам IPv6 распространять такую информацию: "Вот известная мне информация!" На рис. 28.2 представлен лишь частный случай, а именно факт изучения при помощи сообщений RS и RA IPv6-адреса маршрутизатора IPv6.

Еще один факт: маршрутизаторам известен префикс и его длина, используемые в локальном канале связи. На каждом интерфейсе маршрутизатора обычно введена команда `ipv6 address`; она перечисляет длину префикса и достаточно информации для маршрутизатора, чтобы вычислить соответствующий префикс IPv6. Хост может изучить эти подробности, используя обмен сообщениями RS и RA, как показано на рис. 28.3.

Рис. 28.3. Использование сообщений NDP RS/RA для обнаружения префикса и его длины в сети LAN

Кроме того, процесс SLAAC, применяемый хостами для динамического изучения IPv6-адресов, использует информацию о префиксе/длине префикса, полученную от маршрутизатора с использованием сообщений RA и RS. Более подробно этот процесс обсуждается далее.

Обнаружение адресов соседних устройств при помощи сообщений NDP NS и NA

Протокол NDP определяет еще одну пару сообщений запроса и анонса: *запрос соседа* (Neighbor Solicitation — NS) и *анонс соседа* (Neighbor Advertisement — NA). Сообщение NS действует в основном как запрос протокола ARP в случае IPv4, требуя от хоста с конкретным одноадресатным IPv6-адресом отослать ответ назад. Сообщение NA действует как ответ протокола ARP в случае IPv4, сообщая MAC-адрес хоста.

Процесс передачи сообщений NS и NA протекает в общем так же, как и процесс передачи сообщений RS и RA: сообщение NS запрашивает информацию, а сообщение NA предоставляет ее. Наиболее очевидное различие в том, что сообщения RS/RA сосредоточиваются на информации, содержащейся на маршрутизаторах, а сообщения NS/NA — на информации, содержащейся на каждом хосте IPv6. Кроме того, обратите внимание на то, что сообщения NS поступают на многоадресатный адрес опрашиваемого узла, связанный с целевым объектом. Более подробную информацию об этом типе адреса см. в главе 27.

Ключевая тема

Описание сообщений запроса и анонса соседнего устройства

- *Запрос соседа* (Neighbor Solicitation — NS). Просит хост с конкретным IPv6-адресом (целевым адресом) вернуть сообщение NA с указанием его MAC-адреса. Сообщение NS посылают на многоадресатный адрес опрашиваемого узла, связанный с целевым адресом, чтобы сообщение было обработано только теми хостами, последние шесть шестнадцатеричных цифр адреса которых совпадают с запрашиваемым адресом.
- *Анонс соседа* (Neighbor Advertisement — NA). В этом сообщении перечислен адрес отправителя как целевой адрес, а также соответствующий MAC-адрес. Оно отсылается назад по одноадресатному адресу хоста, пославшего первоначальное сообщение NS. В некоторых случаях хост посылает незапрашиваемое сообщение NA, когда сообщение посылают в многоадресатный адрес локальной области видимости FF02::1 (все хосты IPv6).

ВНИМАНИЕ!

В названии протокола обнаружения соседних устройств (Neighbor Discovery Protocol — NDP) слово "сосед" (neighbor) означает тот факт, что устройства находятся на том же канале связи, например, в той же сети VLAN.

На рис. 28.4 приведен пример того, как хост (PC1) использует сообщение NS для изучения MAC-адреса, используемого другим хостом. Сообщения NDP NS и NA заменяют сообщения ARP протокола IPv4 в том смысле, что они позволяет хостам обнаруживать адрес уровня канала связи других хостов IPv6 на том же канале связи. (В протоколе IPv6 хосты на том же канале связи называются просто хостами *на ка-*

нале связи.) В сообщении NS указан целевой одноадресатный адрес IPv6 с подразумеваемым вопросом: "Каков ваш адрес?" В данном примере сообщение NA возвращается первоначальному хосту, задавшему вопрос, и содержит этот адрес. Пример приведен на рис. 28.4.

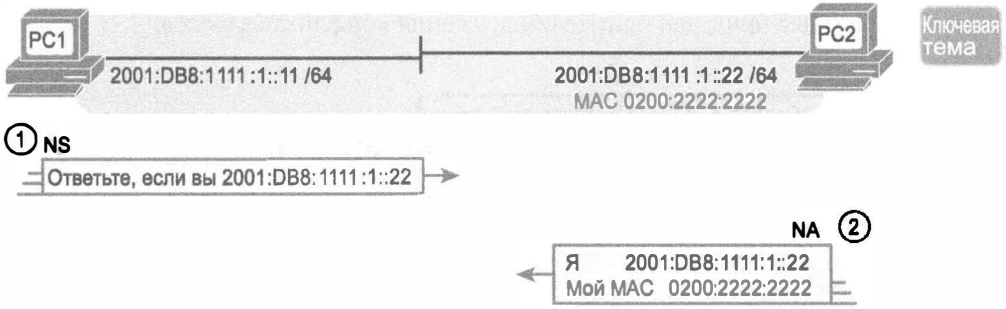


Рис. 28.4. Пример процесса NDP NS/NA по поиску адресов соседей на канале связи

На первом этапе данного конкретного примера компьютер PC1 посылает запрос на поиск MAC-адрес компьютера PC2. Сначала PC1 просматривает свою таблицу соседних устройств NDP, эквивалент кэша ARP протокола IPv4, и не находит MAC-адрес для IPv6-адреса 2001:DB8:1111:1::22. Поэтому на этапе 1 компьютер PC1 посылает сообщение NDP NS на соответствующий многоадресатный адрес опрашиваемого узла 2001:DB8:1111:1::22 или FF02::1:FF00:22. Это сообщение обработают только те хосты IPv6, адреса которых заканчиваются на 00:0022. В результате лишь небольшое подмножество хостов на этом канале связи обработает полученное сообщение NS NDP.

На втором этапе компьютер PC2 реагирует на полученное сообщение NS. Он отсылает назад ответное сообщение NA, содержащее MAC-адрес компьютера PC2. Компьютер PC1 записывает MAC-адрес компьютера PC2 в свою таблицу соседних устройств NDP.

ВНИМАНИЕ!

Чтобы просмотреть таблицу соседних устройств NDP хоста, используйте следующие команды: `netsh interface ipv6 show neighbors` (на Windows); `ip -6 neighbor show` (на Linux); `ndp -an` (на Mac OS).

Обнаружение конфликтов адресов при помощи сообщений NDP NS и NA

Сообщения NDP NS/NA позволяют также хостам выполнить важную проверку, позволяющую избежать использования двойных IPv6-адресов. Прежде чем использовать одноадресатный адрес, хост IPv6 применяет процесс *обнаружения конфликтов адресов* (Duplicate Address Detection — DAD), позволяющий удостовериться в том, что никакой другой хост на том же канале связи уже не использует этот адрес. Если другой хост уже использует этот адрес, первый хост просто не использует его, пока проблема не будет решена.

Термин DAD относится к функции, причем функции, использующей сообщения NDP NA и NS. Проще говоря, хост посылает сообщение NS, но указывает в нем адрес, который собирается использовать. Если такой адрес не используется, ни один из хостов не ответит на сообщение NA. Но если другой хост уже использует этот адрес, то он ответит сообщением NA, засвидетельствовав попытку повторного использования адреса. На рис. 28.5 приведен пример обнаружения конфликта адресов.

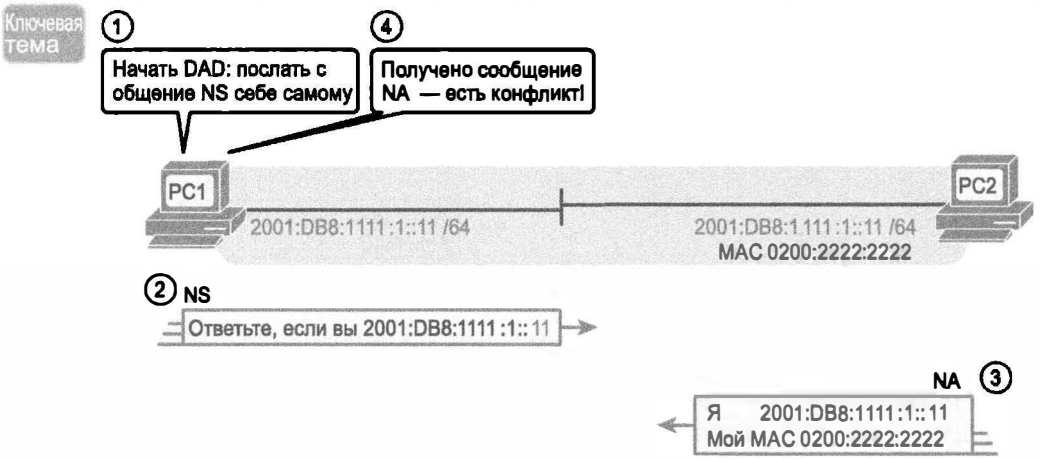


Рис. 28.5. Пример процесса NDP NS/NA по поиску конфликта адресов с соседями на канале связи

На рис. 28.5 приведен пример пытки использования компьютерами PC1 и PC2 одинакового IPv6-адреса. Компьютер PC2 уже использует этот адрес, а компьютер PC1 применяет процесс DAD, прежде чем использовать его. Последовательность действий на рисунке описана ниже.

1. Прежде чем использовать адрес 2001:DB8:1111:1::11, компьютер PC1 применяет процесс DAD.
2. Компьютер PC1 посылает сообщение NS на адрес 2001:DB8:1111:1::11, который он собирается использовать.
3. Компьютер PC2 получает сообщение NS и, поскольку это его адрес, отсылает назад сообщение NA.
4. Компьютер PC1 получает сообщение NA на его собственный IPv6-адрес и понимает, что имеется конфликт адресов.

Хосты используют процесс DAD для проверки всех своих одноадресатных адресов, включая адрес, локальный в пределах канала связи, перед их первым использованием.

Протокол NDP

Эта глава резюмирует некоторые из важнейших функций, осуществляемых протоколом NDP. Протокол NDP выполняет существенно больше функций, чем описано в этой главе, а также допускает добавление других функций, что позволяет совершенствовать его в дальнейшем. Пока можно использовать табл. 28.1 как справочник по четырем обсуждаемым здесь функциям NDP.

Ключевая тема

Таблица 28.1. Функции протокола NDP

Функция	Сообщения протокола	Кто получает информацию	Кто предоставляет информацию	Предоставляемая информация
Обнаружение маршрутизатора	RS и RA	Любой хост IPv6	Любой маршрутизатор IPv6	Локальный для канала связи IPv6-адрес маршрутизатора
Обнаружение префикса/длины	RS и RA	Любой хост IPv6	Любой маршрутизатор IPv6	Префикс (префиксы) и связанные с ним длины префикса, используемые на локальном канале связи
Обнаружение соседнего устройства	NS и NA	Любой хост IPv6	Любой хост IPv6	Используемый соседом адрес уровня канала связи (например, MAC-адрес)
Обнаружение конфликта адресов	NS и NA	Любой хост IPv6	Любой хост IPv6	Простое подтверждение, не используется ли уже одноадресатный адрес

Динамическая настройка параметров IPv6 на хосте

На момент создания протокола IPv6, в середине 1990-х годов, протокол IPv4 насчитывал два десятилетия практического опыта работы. Этот опыт засвидетельствовал потребность хостов в динамическом изучении своих параметров IPv4, включая IPv4-адрес хоста. На момент создания протокола IPv6 протокол DHCP для IPv4 уже стал предпочтительным решением, позволяющим хостам динамически изучать свои IPv4-адреса и другие параметры.

Протокол DHCP хорошо работал с протоколом IPv4, поэтому создание версии протокола DHCP для протокола IPv6 (DHCPv6) имело смысл. Хотя протокол DHCP имел много преимуществ, но он имел и один недостаток — потребность в сервере DHCP, хранящем информацию о каждом хосте (клиенте) и его адресе. Разработчики протокола IPv6 хотели создать альтернативный инструмент динамического присвоения адресов, не требующий сервера. Ответом была *автоматическая настройка адреса* (Stateless Address Autoconfiguration — SLAAC).

В этом разделе сначала рассмотрим протокол DHCPv6, а затем процесс SLAAC.

Динамическая настройка с использованием протоколов DHCP с фиксацией состояния и NDP

Протокол DHCP для IPv6 (DHCPv6) позволяет хосту IPv6 изучать параметры конфигурации IPv6, используя те же общие концепции, что и протокол DHCP для IPv4. Хост обменивается сообщениями с сервером DHCP, а сервер снабжает хост информацией о конфигурации, включая резервирование IPv6-адреса, информацию о длине префикса и адресе сервера DNS.

ВНИМАНИЕ!

Фактически версия протокола DHCP не является шестой; его название только заканчивается на "v6", означая поддержку протокола IPv6.

Более конкретно, протокол DHCPv6 с фиксацией состояния во многом работает как уже знакомый протокол DHCP для IPv4.

Ключевая тема

Подобия между протоколом DHCP для IPv4 и протоколом DHCP с фиксацией состояния для IPv6

- Клиенты DHCP в сети LAN посылают сообщения, передаваемые только в локальной сети, надеясь найти сервер DHCP.
- Если сервер DHCP находится в той же сети LAN, что и клиент, то клиент и сервер смогут обмениваться сообщениями DHCP непосредственно, не нуждаясь в помощи маршрутизатора.
- Если сервер DHCP и клиент находятся на разных каналах связи, то для перенаправления сообщений DHCP клиент и сервер полагаются на маршрутизатор.
- Маршрутизатор, перенаправляющий сообщения между каналами связи на сервер в дистанционной подсети, следует настроить как *агент пересылки DHCP* (DHCP Relay Agent), знающий IPv6-адрес сервера DHCP.
- Конфигурация серверов содержит пулы адресов для каждой подсети, для которой сервер резервирует адреса.
- Серверы позволяют клиенту резервировать IP-адреса из пула адресов для подсети клиента; резервирование осуществляется на заданный период времени (обычно дни или недели).
- Сервер отслеживает информацию о состоянии, а именно идентификатор клиента (зачастую на основании MAC-адреса) и адрес, зарезервированный клиентом в настоящее время.

Есть два главных направления применения протокола DHCPv6: с фиксацией состояния и без таковой. Протокол DHCPv6 с фиксацией состояния работает скорее как модель протокола DHCPv4, особенно в отношении последнего элемента в списке. Сервер DHCPv6 с фиксацией состояния отслеживает информацию о клиентах и зарезервированных ими IPv6-адресах; благодаря тому факту, что сервер знает информацию о конкретном клиенте, называемую информацией о состоянии, такой сервер DHCP является сервером с фиксацией состояния.

Серверы DHCP без фиксации состояния не отслеживают информацию о клиенте. В следующем разделе, "Использование автоматической настройки адреса", будет показано, что у серверов DHCPv6 без фиксации состояния есть важная роль, когда компания решает использовать SLAAC.

Различия между протоколами DHCPv6 и DHCPv4

Хотя у протокола DHCPv6 с фиксацией состояния много сходств с протоколом DHCPv4, между ними есть также много отличий. Одно из главных отличий приведено на рис. 28.6: протокол DHCPv6 с фиксацией состояния не предоставляет клиенту информацию о стандартном маршрутизаторе. Вместо этого хост клиента использует встроенный протокол NDP, позволяющий получать IPv6-адреса маршрутизаторов непосредственно от локальных маршрутизаторов.

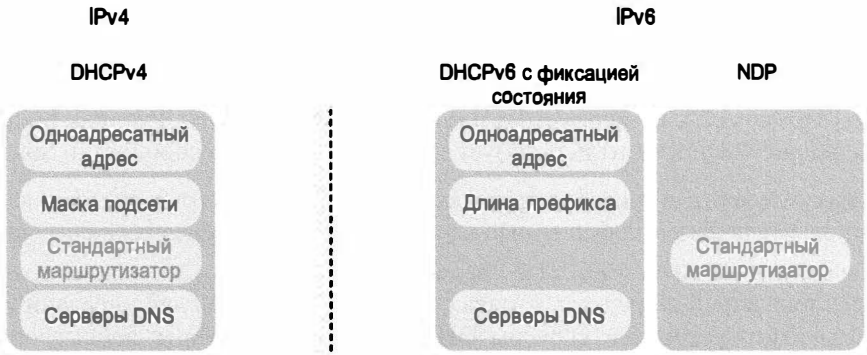


Рис. 28.6. Источники параметров IPv6 при использовании протокола DHCP с фиксацией состояния

Кроме того, сообщения протокола DHCPv6 изменены и дополнены новыми полями, чтобы использовались пакеты IPv6 вместо пакетов IPv4. На рис. 28.7 представлены названия сообщений Solicit (Требование), Advertise (Анонс), Request (Запрос) и Reply (Ответ) протокола DHCPv6, заменяющие сообщения DORA (Discover, Offer, Request and Acknowledgment — Обнаружить, Предложение, Запрос и Подтверждение) протокола DHCPv4.

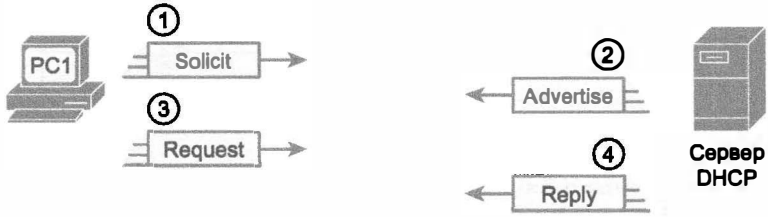


Рис. 28.7. Четыре сообщения протокола DHCPv6 с фиксацией состояния, передаваемые между клиентом и сервером

Четыре сообщения DHCPv6 работают соответствующими парами, как и подобные сообщения DHCPv4. Сообщения Solicit и Advertise участвуют в процессе поиска клиентом IPv6-адреса сервера DHCPv6. Клиент запрашивает (сообщение Solicit), а сервер анонсирует адрес (и другие параметры конфигурации) при помощи сообщения Advertise. Сообщение Request позволяют клиенту запросить резервируемый адрес, а сервер подтверждает резервирование сообщением Reply.

Агенты пересылки DHCPv6

На предприятиях, решивших использовать сервер DHCPv6 с фиксацией состояния, он зачастую находится на центральной площадке, далеко от большинства использующих его клиентов DHCPv6. В таких случаях локальный маршрутизатор на каждой площадке должен действовать как агент пересылки DHCP.

Концепция пересылки DHCPv6 напоминает работу ретрансляторов DHCPv4, обсуждавшихся в главе 18. Клиент посылает сообщение, передаваемое только по локальной сети. Затем маршрутизатор изменяет IP-адрес отправителя и получателя, перенаправляя пакет на сервер DHCP. Когда сервер посылает ответ, он фактически

направляет его на адрес маршрутизатора (агент пересылки), который изменяет адреса и этого пакета.

Отличие протокола IPv6 становятся более очевидными, если рассмотреть некоторые из IPv6-адресов, используемых в таких сообщениях DHCPv6, как Solicit. Как показано на рис. 28.8, клиент использует следующие адреса в сообщении Solicit.

- **Адрес отправителя локален в пределах канала связи.** Клиент использует как адрес отправителя пакета собственный адрес, локальный в пределах канала связи.
- **Адрес получателя — FF02::1:2 "все агенты DHCP".** Это многоадресатный адрес, локальный в пределах канала связи, используемый для передачи пакетов на два типа устройств: серверы DHCP и маршрутизаторы, действующие как агенты пересылки DHCP.

Если адрес получателя имеет область видимости, локальную в пределах канала связи, то посланное хостом сообщение Solicit передавалось бы только в пределах локальной сети. На рис. 28.8 представлены некоторые из частностей того, как маршрутизатор R1, выступая в качестве агента пересылки DHCPv6, позволяет таким клиентам DHCPv6, как хост, доставлять пакеты DHCPv6 серверу DHCPv6.

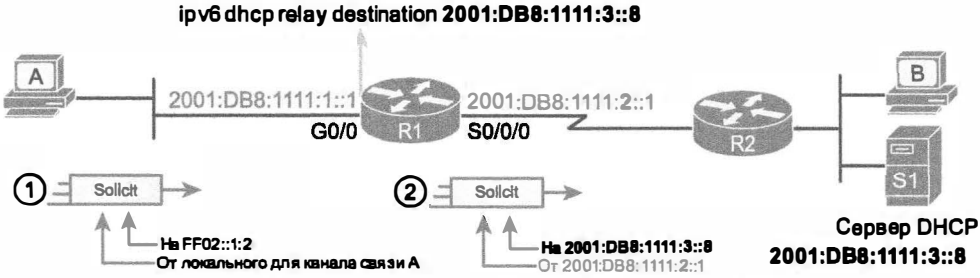


Рис. 28.8. Агент пересылки DHCPv6 и IPv6-адреса DHCP

Рассмотрим сначала этап 1, когда хост A, являющийся клиентом DHCPv6, создает и посылает сообщение DHCPv6 Solicit. Сообщение передается с адреса хоста A, локального в пределах канала связи, на многоадресатный адрес FF02::1:2 (все агенты DHCP). При многоадресатном адресе получателя с локальной областью видимости канала связи посланное хостом сообщение Solicit передается только в локальной сети.

Этап 2 демонстрирует результаты работы маршрутизатора R1 как агента пересылки DHCPv6. Маршрутизатор R1 прослушивает входящие сообщения DHCPv6, посланные на адрес FF02::1:2, и обрабатывает сообщение, посланное хостом A. Маршрутизатор R1 изменяет IPv6-адрес получателя пакета, чтобы он соответствовал адресу сервера DHCPv6. Он также изменяет исходный IPv6-адрес, чтобы он был одним из IPv6-адресов маршрутизатора R1. Изначально, как исходящий IPv6-адрес, маршрутизатор R1 использует адрес своего исходящего интерфейса (S0/0/0), что немного отличается от агента пересылки DHCPv4. Затем маршрутизатор R1 перенаправляют сообщение Solicit на сервер.

Возвращение сообщения DHCPv6 от сервера клиенту на рисунке не показано, оно поступает сначала на IPv6-адрес маршрутизатора агента пересылки, в данном случае 2001:DB8:1111:2::1. Затем агент пересылки преобразует адрес получателя этих

сообщений DHCPv6 и перенаправляют их на адрес клиента, локальный в пределах канала связи.

Пример 28.1 демонстрирует настройку агента пересылки DHCPv6 для маршрутизатора R1, согласно рис. 28.8. В верхней части примера показана подкоманда интерфейса `ipv6 dhcp relay` с IPv6-адресом сервера DHCPv6. В нижней части представлен вывод команды `show ipv6 interface`, подтверждающий, что маршрутизатор R1 теперь прослушивает многоадресатные сообщения, посланные на многоадресатный адрес FF02::1:2 (все агенты DHCP).

Пример 28.1. Настройка маршрутизатора R1 для поддержки дистанционного сервера DHCPv6

```
interface GigabitEthernet0/0
  ipv6 dhcp relay destination 2001:DB8:1111:3::8

R1# show ipv6 interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::FF:FE00:1
  No Virtual link-local address(es):
  Description: to SW1 port F0/1
  Global unicast address(es):
    2001:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64 [EUI]
Joined group address(es):
  FF02::1
  FF02::2
  FF02::A
  FF02::1:2
  FF02::1:FF00:1
! Строки пропущены для краткости
```

Использование автоматической настройки адреса

У протокола DHCPv4 с фиксацией состояния те же характеристики, что и у его новой версии, DHCPv6 с фиксацией состояния, и несколько тех же проблем. Кто-то должен настраивать, администрировать и управлять серверами DHCP. Настройка подразумевает задание диапазона IP-адресов для каждой подсети. Затем, когда хост (клиент) зарезервирует адрес, сервер отслеживает адреса, используемые клиентами. Все эти функции хорошо работают, но уверенность в работоспособности сервера DHCP с фиксацией состояния требует знания и внимания от штата IT.

Автоматическая настройка адреса IPv6 (Stateless Address Autoconfiguration — SLAAC) обеспечивает альтернативный метод динамического присвоения IPv6-адресов, не нуждающийся в сервере с фиксацией состояния. Другими словами, SLAAC не требуется сервер, присваивающий или резервирующий IPv6-адреса, штат IT не обязан предварительно настраивать данные о подсетях и не требует, чтобы сервер отслеживал устройства, использующие IPv6-адреса.

Термин *SLAAC* относится к обеим частям способа изучения хостом одного из параметров IPv6 (его IPv6-адрес), а также к общему процессу изучения всех четырех ключевых параметров хостов IPv6 (адреса, длины префикса, адресов стандартного маршрутизатора и сервера DNS). Следующая тема начинается с рассмотрения решаемых SLAAC задач, связанных с IPv6-адресом. Затем будет рассмотрен общий процесс, используемый SLAAC, для поиска всех четырех параметров хоста, используемых протоколом NDP, а также сервером DHCP без фиксации состояния.

Создание IPv6-адреса с использованием SLAAC

При использовании SLAAC хост не резервирует свой IPv6-адрес и даже не изучает его. Вместо этого он изучает часть адреса (префикс), а затем вычисляет остальную часть собственного IPv6-адреса. А именно: для выбора собственного IPv6-адреса при помощи SLAAC хост использует приведенную ниже последовательность действий.

Ключевая тема

Этапы создания хостом своего IPv6-адреса с использованием SLAAC

1. Выяснение при помощи сообщений NDP RS/RA у любого маршрутизатора префикса IPv6, используемого на канале связи.
2. Вычисление собственного IPv6-адреса на основании значения идентификатора интерфейса и только что полученного префикса IPv6.
3. Прежде чем использовать адрес, хост использует процесс DAD, чтобы удостовериться в том, что никакой другой хост уже не использует тот же адрес.

Рис. 28.9 резюмирует первые два этапа, уделяя внимание двум наиболее распространенным способам получения адреса хостом. Хосты могут использовать правила EUI-64, как обсуждалось в главе 27. Но хост вполне может использовать процесс, использующий случайное число.

Ключевая тема

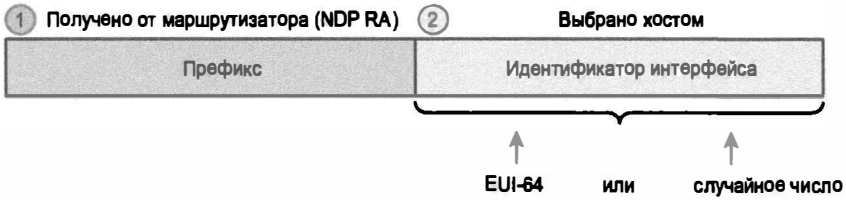


Рис. 28.9. Создание IPv6-адреса хоста с использованием SLAAC

ВНИМАНИЕ!

Операционная система Microsoft OS предпочитает выбирать идентификатор интерфейса случайно, с регулярной сменой клиентами их значений в целях безопасности.

Объединение SLAAC с протоколами NDP и DHCP без фиксации состояния

Используя процесс SLAAC, хост фактически применяет три разных инструментальных средства для поиска своих четырех параметров IPv6 (рис. 28.10). Сам процесс SLAAC сосредоточивается только на IPv6-адресе. Затем хост использует сообщения NDP для получения длины префикса и IPv6-адресов маршрутизаторов, доступных на канале связи. И наконец, хост использует протокол DHCP без фиксации состояния, чтобы узнать IPv6-адреса любых серверов DNS.

Протокол DHCP без фиксации состояния решает последнюю часть этой проблемы, также используя SLAAC. Хост должен знать IPv6-адреса серверов DNS. Решение? Использование протокола DHCPv6. Но хост, действуя как клиент DHCPv6, запрашивает у сервера только адреса серверов DNS и не резервирует IPv6-адреса.

Так почему эту службу назвали сервером *DHCPv6 без фиксации состояния*? При использовании сервера DHCP без фиксации состояния у сетевого инженера гораздо меньше работы по настройке и администрированию. Сервер DHCPv6 без фиксации состояния:

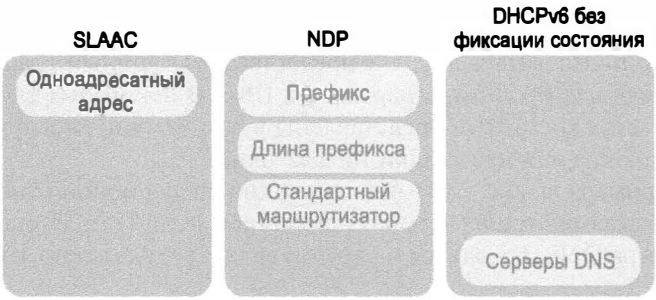


Рис. 28.10. Источники определенных параметров IPv6 при использовании SLAAC

- нуждается только в простой настройке, а именно небольшом количестве адресов для серверов DNS, и ничего больше;
- нет никакой необходимости в настройке подсетей: никаких списков подсетей, пулов адресов подсетей, списков исключенных адресов в подсетях и никаких длин префикса в подсетях;
- нет никакой необходимости отслеживать информацию о состоянии зарезервированных адресов DHCP (какое устройство какой IPv6-адрес резервирует), поскольку сервер не предоставляет адресов никаким клиентам.

В табл. 28.2 подведен краткий итог сравнения ключевых различий между серверами DHCP с фиксацией состояния и без нее.

Таблица 28.2. Сравнение служб DHCPv6 с фиксацией состояния и без

Средство	С фиксацией состояния	Без фиксации состояния
Помнит IPv6-адреса (информацию о состоянии) клиентов, сделавших запрос	Да	Нет
Резервирует IPv6-адрес для клиента	Да	Нет
Предоставляет список адресов серверов DNS	Да	Да
Общепринят в SLAAC	Нет	Да

Проверка подключения хоста IPv6

В этом заключительном разделе рассматривается несколько команд, позволяющих проверить параметры хостов. В частности, исследуются параметры хоста IPv6, а затем наиболее популярные команды, позволяющие проверить, способен ли хост посылать пакеты: `ping` и `traceroute`.

Обратите внимание, что в этом разделе приведены некоторые из команд для разных OS хоста. Как обычно, цель демонстрации команд хоста — дать общее представление об информации, которую можно просмотреть на хосте. Однако имейте в виду, что в данной и других главах не было попытки продемонстрировать все варианты каждой команды на каждой OS; их задача лишь в том, чтобы укрепить понимание концепций, рассмотренных ранее в главе.

Проверка подключения хостов IPv6

Большинство пользовательских OS предоставляет удобный графический интерфейс пользователя для просмотра параметры IPv6. В некоторых случаях все четыре ключевых параметра хоста IPv6 представлены в том же окне, а в других приходится перейти в другие окна или на вкладки того же окна.

В качестве примера на рис. 28.11 показано окно операционной системы Mac OS X, в котором отображаются три из четырех параметров хоста IPv6. Отсутствующий параметр, адрес сервера DNS, находится на другой вкладке (как можно заметить в верхней части окна).

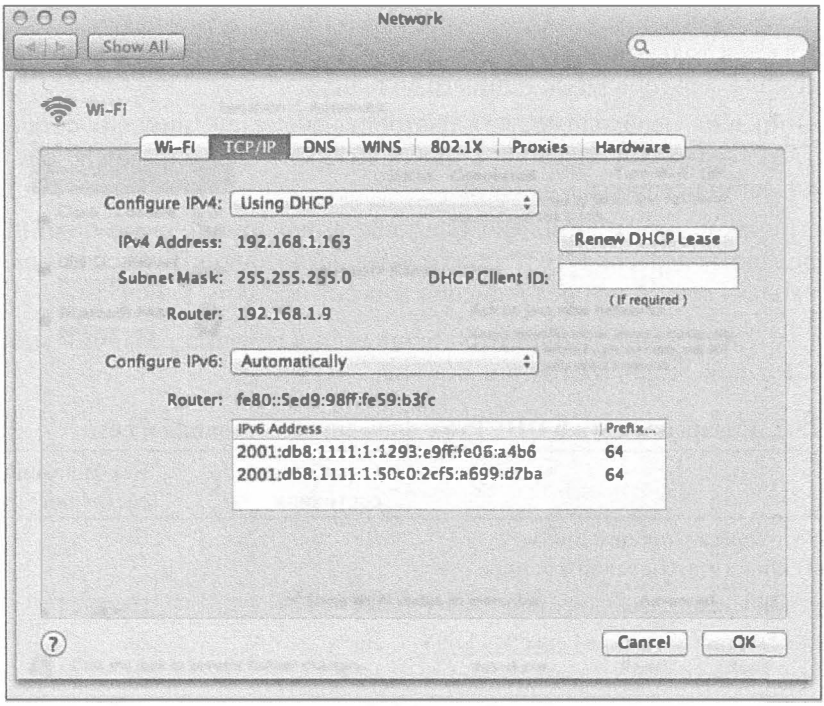


Рис. 28.11. Три параметра IPv6 в операционной системе Mac OS

Уделите минуту изучению деталей, представленных на рис. 28.11. Сверху показаны параметры IPv4, полученные по протоколу DHCP. В нижней половине окна показаны параметры IPv6, полученные автоматически, это значит, что хост будет использовать либо протокол DHCP с фиксацией состояния, либо SLAAC. В данном случае хост использовал процесс SLAAC для присвоения себе двух IPv6-адресов в той же подсети 2001:DB8:1111:1::/64, одного с использованием правил EUI-64 и одного со случайным идентификатором интерфейса. (Обратите внимание, что логика хоста IPv6 имеет много деталей, не обсуждаемых в этой главе, включая причины, почему хост мог бы использовать два адреса, а не один.)

Хосты поддерживают также набор команд для проверки этой информации. Многие операционные системы используют для параметров IPv6 знакомые команды: `ipconfig` на OS Windows и `ifconfig` на Linux и Mac OS. Пример 28.2 демон-

стрирует команду `ifconfig` на том же компьютере Mac, окно которого представлено на рис. 28.11. В частности, если рассмотреть два выделенных поля, то можно заметить идентификатор интерфейса EUI-64, полученный на основании MAC-адреса этого хоста.

Пример 28.2. Пример команды `ifconfig` в операционной системе Mac

```
WOair$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
     ether 10:93:e9:06:a4:b6
     inet6 fe80::1293:e9ff:fe06:a4b6%en0 prefixlen 64 scopeid 0x4
     inet 192.168.1.163 netmask 0xffffff00 broadcast 192.168.1.255
     inet6 2001:db8:1111:1:1293:e9ff:fe06:a4b6 prefixlen 64 autoconf
     inet6 2001:db8:1111:1:50c0:2cf5:a699:d7ba prefixlen 64 autoconf
     temporary
     media: autoselect
     status: active
```

Кроме простой проверки четырех ключевых параметров IPv6, установка нового хоста также требует проверки наличия у хоста подключения к остальной части объединенной сети. Для этого обычно используются команды `ping` и `traceroute`.

Что касается самих команд, то некоторые операционные системы (особенно разновидности Microsoft Windows, маршрутизаторы и коммутаторы Cisco) позволяют использовать те же команды `ping` и `traceroute`, используемые протоколом IPv4. Некоторые другие операционные системы требуют других команд, таких как `ping6` и `traceroute6`, используемых операционными системами Mac OS и Linux. (Следующие примеры демонстрируют оба варианта.)

Что касается вывода команд `ping` и `traceroute`, то большинству пользователей он хорошо известен по версии IPv4, и при переходе на версию IPv6 не требует ни изучения, ни тренировки. Изменения в выводе, по сравнению с эквивалентом IPv4, главным образом заключаются в применении IPv6-адресов. Для сравнения в примерах 28.3 и 28.4 демонстрируется типичный вывод для объединенной сети, изображенной на рис. 28.12.

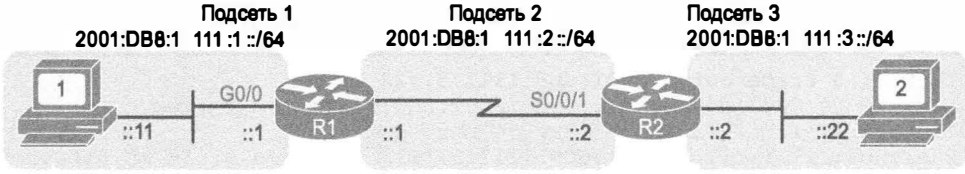


Рис. 28.12. Объединенная сеть IPv6 для примеров команд `traceroute` и `ping`

Пример 28.3 демонстрирует три команды `ping`, отданные на хосте PC1 под операционной системой Linux. (На Linux прежние команды заменены командами `ping6` и `traceroute6`.) Первые две команды представляют эхо-запросы IPv6: первая — для IPv6-адреса LAN маршрутизатора R1, вторая — для IPv6-адреса компьютера PC2. Заключительная команда демонстрирует для сравнения команду `ping` IPv4.

Ключевая тема **Пример 28.3. Команда ping6, отданная на компьютере PC1 для хостов R1 и PC2**

```
Master@PC1:$ ping6 2001:db8:1111:1::1
PING 2001:db8:1111:1::1 (2001:db8:1111:1::1) 56 data bytes
64 bytes from 2001:db8:1111:1::11: icmp_seq=1 ttl=64 time=1.26 ms
64 bytes from 2001:db8:1111:1::11: icmp_seq=2 ttl=64 time=1.15 ms
^C
--- 2001:db8:1111:1::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001 ms
rtt min/avg/max/mdev = 1.156/1.210/1.263/0.062 ms
Master@PC1:$ ping6 2001:db8:1111:3::22
PING 2001:db8:1111:3::22 ( 2001:db8:1111:3::22) 56 data bytes
64 bytes from 2001:db8:1111:3::22: icmp_seq=1 ttl=64 time=2.33 ms
64 bytes from 2001:db8:1111:3::22: icmp_seq=2 ttl=64 time=2.59 ms
64 bytes from 2001:db8:1111:3::22: icmp_seq=3 ttl=64 time=2.03 ms
^C
--- 2001:db8:1111:3::22 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003 ms
rtt min/avg/max/mdev = 2.039/2.321/2.591/0.225 ms

! Команда ping IPv4 для сравнения
Master@PC1:$ ping 10.1.3.22
PING 10.1.3.22 (10.1.3.22) 56 data bytes
64 bytes from 10.1.3.22: icmp_seq=1 ttl=64 time=2.45 ms
64 bytes from 10.1.3.22: icmp_seq=2 ttl=64 time=2.55 ms
64 bytes from 10.1.3.22: icmp_seq=3 ttl=64 time=2.14 ms
^C
--- 10.1.3.22 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2014 ms
rtt min/avg/max/mdev = 2.04/2.318/2.604/0.224 ms
```

Пример 28.4 демонстрирует команду traceroute6 на компьютере PC1, выводящую маршрут к хосту PC2. Пример отражает стиль вывода для большинства команд traceroute IPv4, кроме очевидного отличия IPv6-адресов. Обратите внимание, что в выводе указан IPv6-адрес интерфейса G0/0 маршрутизатора R1, затем IPv6-адрес интерфейса S0/0/1 маршрутизатора R2 и адрес хоста PC2, завершающий вывод.

Пример 28.4. Команда traceroute6, отданная на компьютере PC1 для хоста PC2

```
Master@PC1:$ traceroute6 2001:db8:1111:3::22
traceroute to 2001:db8:1111:3::22 (2001:db8:1111:3::22) from
2001:db8:1111:1::11, 30 hops max, 24 byte packets
1  2001:db8:1111:1::1  (2001:db8:1111:1::1)  0.794 ms 0.648 ms 0.604 ms
2  2001:db8:1111:2::2  (2001:db8:1111:2::2)  1.606 ms 1.49 ms  1.497 ms
3  2001:db8:1111:3::22 (2001:db8:1111:3::22) 2.038 ms 1.911 ms 1.899 ms
```

Проверка подключения хоста от соседних маршрутизаторов

Некоторые средства проверки маршрутизаторов IPv6 используют те же команды, что и IPv4, но иногда с заменой ip на ipv6. В некоторых случаях, особенно для функций, отсутствующих в протоколе IPv4 или измененных весьма незначительно, маршрутизаторы поддерживают совершенно новые команды. В данном разделе рассматривается несколько команд маршрутизатора, полезных при проверке подключения хоста IPv6, включая как старые, так и некоторые новые.

Сначала рассмотрим знакомые команды. Маршрутизаторы и коммутаторы Cisco поддерживают команды `ping` и `traceroute` с теми же основными характеристиками, что и для протокола IPv4. Для стандартных версий команд передается адрес IPv4 или IPv6. Для расширенных версий этих команд первый запрос в приглашении просит указать тип протокола `ipv6`, а не использовать стандартное значение `ip`.

Как обычно, поможет пример, особенно для расширенных команд. Пример 28.5 начинается с расширенной команды `ping` протокола IPv6, отданной на компьютере PC1 для хоста PC2 с использованием интерфейса G0/0 маршрутизатора R1 как отправителя пакетов. Вторая команда — это стандартная команда `traceroute` протокола IPv6, отданная на маршрутизаторе R1 для хоста PC2.

Пример 28.5. Расширенная команда `ping` и стандартная команда `traceroute` на маршрутизаторе R1

```
R1# ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:1111:3::22
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Source address or interface: GigabitEthernet0/0
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1111:3::22, timeout is 2
seconds:
Packet sent with a source address of 2001:DB8:1111:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/4 ms

R1# traceroute 2001:db8:1111:3::22
Type escape sequence to abort.
Tracing the route to 2001:DB8:1111:3::22

    1 2001:DB8:1111:2::2 4 msec 0 msec 0 msec
    2 2001:DB8:1111:3::22 0 msec 4 msec 0 msec
```

Еще один способ проверки параметров хоста с маршрутизатора подразумевает просмотр таблицы соседних устройств маршрутизатора. Все хосты IPv6, включая маршрутизаторы, хранят таблицу соседних устройств IPv6: список всех IPv6-адресов и соответствующих им MAC-адресов соседних устройств. В основном эта таблица заменяет таблицу ARP протокола IPv4 и содержит данные, изученные в сообщениях NDP NA и NS.

Один из способов проверки доступности соседнего хоста — удостовериться в том, вернет ли он сообщение NDP NA, когда маршрутизатор пошлет ему сообщение NDP NS (чтобы обнаружить MAC-адрес хоста). Для этого маршрутизатор может очистить

свою таблицу соседних устройств (команда `clear ipv6 neighbor`), а затем применить команду `ping` для хоста на соответствующем интерфейсе. Сначала маршрутизатор должен послать сообщение NDP NS, а хост должен послать назад сообщение NDP NA. Если маршрутизатор отображает MAC-адрес этого хоста в таблице соседних устройств, значит, хост ответил на сообщение NDP NS. Пример 28.6 демонстрирует вывод таблицы соседних устройств IPv6 маршрутизатора R2, согласно рис. 28.13, с использованием команды `show ipv6 neighbors`.

Пример 28.6. Команда `show ipv6 neighbors` на маршрутизаторе R2

```
R2# show ipv6 neighbors
IPv6 Address                Age Link-layer Addr State Interface
FE80::11FF:FE11:1111          0 0200.1111.1111 STALE Gi0/0
FE80::22FF:FE22:2222          1 0200.2222.2222 STALE Gi0/0
2001:DB8:1111:3::22           0 0200.2222.2222 REACH Gi0/0
FE80::FF:FE00:3333            1 0200.0000.3333 DELAY Gi0/0
2001:DB8:1111:3::33           0 0200.1111.1111 REACH Gi0/0
2001:DB8:1111:3::3            0 0200.0000.3333 REACH Gi0/0
```

И наконец, маршрутизаторы могут также вывести информацию о доступных маршрутизаторах в подсети LAN. Напомню, что маршрутизаторы посылают сообщения NDP RA, чтобы объявить о своей готовности действовать в качестве маршрутизатора IPv6 в конкретной подсети LAN. Маршрутизаторы Cisco отслеживают сообщения RA, передаваемые другими маршрутизаторами (маршрутизаторы сами периодически посылают сообщения RA). Команда `show ipv6 routers` отображает все остальные маршрутизаторы, но не локальный маршрутизатор.

В качестве примера рассмотрим топологию на рис. 28.13. R1 — единственный маршрутизатор IPv6 в сети LAN слева, поэтому маршрутизатор R1 не прослушивает сообщения RA от других маршрутизаторов в той же подсети LAN. Но маршрутизаторы R2 и R3 подключены к той же подсети и слышат сообщения NDP RA друг от друга. Пример 28.7 демонстрирует вывод команды `show ipv6 routers` на маршрутизаторе R1 (без перечисленных маршрутизаторов) и R2 (с одним перечисленным маршрутизатором) для пользы сравнения.

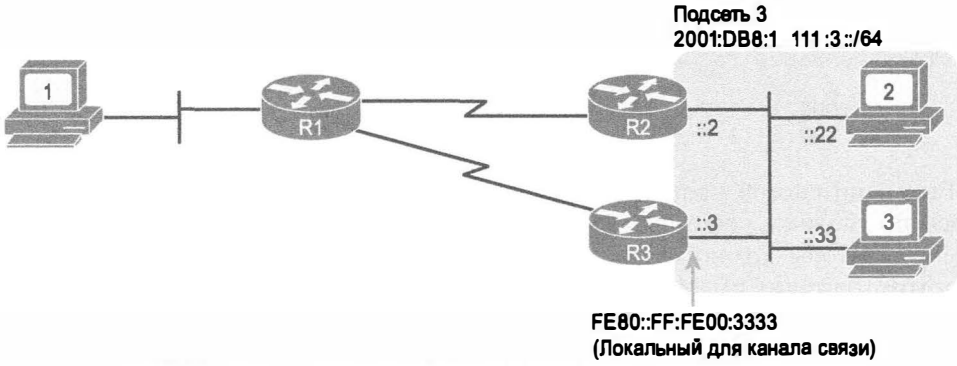


Рис. 28.13. Пример объединенной сети IPv6 с двумя маршрутизаторами на том же канале связи (VLAN)

Пример 28.7. Вывод всех маршрутизаторов при помощи команды `show ipv6 routers`

```
! Эта команда на маршрутизаторе R1 никаких маршрутизаторов не выводит
R1# show ipv6 routers
R1#
! ========================================
! Следующая команда на маршрутизаторе R2 выводит один маршрутизатор (R3)
R2# show ipv6 routers
Router FE80::FF:FE00:3333 on GigabitEthernet0/0, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
  HomeAgentFlag=0, Preference=Medium
  Reachable time 0 (unspecified), Retransmit time 0 (unspecified)
  Prefix 2001:DB8:1111:3::/64 onlink autoconfig
    Valid lifetime 2592000, preferred lifetime 604800
```

И наконец, последний момент, связанный с командами на самих хостах: хост вполне может перечислить собственную информацию NDP. Интересно, что большинство хостов выводят таблицу соседних устройств, а затем только отмечают, какие записи относятся к маршрутизаторам (уже пославшим сообщения NDP RA в некий момент).

Пример 28.8 демонстрирует на сей раз хост под операционной системой Mac OS. Первая из двух выделенных записей со значением "R" в поле флагов ("Flgs") относится к маршрутизатору, пославшему прежде сообщение RA, чтобы объявить о себе. Вторая выделенная запись относится к хосту, поэтому символ "R" приведен не в поле флагов.

Пример 28.8. Пример таблицы соседних устройств NDP на Мак OS

```
WOAir$ ndp -an
Neighbor                Linklayer Address      Netif Expire    St Flgs Prbs
::1                     (incomplete)           lo0 permanent   R
2001:db8:1111:1::1      5c:d9:98:59:b3:fc      en0 1s          D  R
2001:db8:1111:1:1293:e9ff:fe06:a4b6 10:93:e9:6:a4:b6 en0 5s  R
```

Обзор

Резюме

- Протокол обнаружения соседних устройств (NDP) определяет несколько функций, связанных с адресацией IPv6, следующим образом.
  - **SLAAC**. При использовании автоматической настройки адреса (SLAAC) хост использует сообщения NDP для изучения первой части его адреса и длины префикса.
  - **Поиск маршрутизатора**. Используя сообщения NDP, хосты изучают IPv6-адреса доступных маршрутизаторов IPv6 в той же подсети.
  - **Обнаружение конфликта адресов**. Независимо от того, как хост устанавливает или изучает свой IPv6-адрес, он не применяет его, пока не убедится, что никакой другой хост не использует тот же адрес. Как хост решает эту проблему? Используя сообщения NDP, конечно, и процесс обнаружения конфликта адресов (DAD).
  - **Обнаружение MAC-адресов соседних хостов**. После прохождения процесса DAD и начала использования своего IPv6-адреса LAN-ориентированный хост должен изучить MAC-адреса других хостов в той же подсети. Протокол NDP заменяет протокол ARP стека IPv4, поддерживая сообщения, заменяющие сообщения запроса и ответа протокола ARP.
- Собственный IPv6-адрес компьютера — это, как правило, глобальный одноадресатный или уникальный локальный одноадресатный адрес, используемый компьютером при обращении к серверам DNS. Но поскольку стандартный маршрутизатор должен быть доступен локально, ему обычно присваивается адрес, локальный в пределах канала связи маршрутизатора.
- Протокол NDP определяет соответствующую пару сообщений, позволяющих хосту динамически обнаруживать все потенциальные стандартные маршрутизаторы, находящиеся на том же канале связи. Упрощенно процесс сводится к передаче хостом многоадресатного сообщения с запросом "маршрутизаторы, расскажите мне о себе" и ответом маршрутизаторов. Возможны следующие сообщения.
  - **Запрос на получение информации о наличии маршрутизатора** (RS). Сообщение, передаваемое на многоадресатный адрес локальной области видимости FF02::2 ("все маршрутизаторы IPv6"), чтобы опросить все маршрутизаторы только на локальном канале связи и попросить их идентифицировать себя.
  - **Анонс маршрутизатора** (RA). Сообщение, передаваемое маршрутизатором и содержащее множество фактов, включая локальный для канала связи IPv6-адрес маршрутизатора. Не будучи запрошенным, оно передается на многоадресатный адрес локальной области видимости FF02::1 ("все хосты IPv6"). Будучи послано в ответ на сообщение RS, оно передается либо назад, на одноадресатный адрес пославшего запрос хоста, либо на адрес FF02::1 ("все хосты IPv6").

- Протокол NDP определяет еще одну пару сообщений запроса и анонса: запрос соседа (NS) и анонс соседа (NA). Сообщение NS действует в основном как запрос протокола ARP в случае IPv4, требуя от хоста с конкретным одноадресатным IPv6-адресом отослать ответ назад. Сообщение NA действует как ответ протокола ARP в случае IPv4, сообщая MAC-адрес хоста.
- Протокол DHCP для IPv6 (DHCPv6) позволяет хосту IPv6 изучать параметры конфигурации IPv6, используя те же общие концепции, что и протокол DHCP для IPv4. Хост обменивается сообщениями с сервером DHCP, а сервер снабжает хост информацией о конфигурации, включая резервирование IPv6-адреса, информацию о длине префикса и адресе сервера DNS.
- Протокол DHCPv6 с фиксацией состояния во многом работает как уже знакомый протокол DHCP для IPv4.
  - Клиенты DHCP в сети LAN посылают сообщения, передаваемые только в локальной сети, надеясь найти сервер DHCP.
  - Если сервер DHCP находится в той же сети LAN, что и клиент, то клиент и сервер смогут обмениваться сообщениями DHCP непосредственно, не нуждаясь в помощи маршрутизатора.
  - Если сервер DHCP и клиент находятся на разных каналах связи, то для перенаправления сообщений DHCP клиент и сервер полагаются на маршрутизатор.
  - Маршрутизатор, перенаправляющий сообщения между каналами связи на сервер в дистанционной подсети, следует настроить как агент пересылки DHCP (DHCP Relay Agent), знающий IPv6-адрес сервера DHCP.
  - Конфигурация серверов содержит пулы адресов для каждой подсети, для которой сервер резервирует адреса.
  - Серверы позволяют клиенту резервировать IP-адреса из пула адресов для подсети клиента; резервирование осуществляется на заданный период времени (обычно дни или недели).
  - Сервер отслеживает информацию о состоянии, а именно идентификатор клиента (зачастую на основании MAC-адреса) и адрес, зарезервированный клиентом в настоящее время.
- При использовании SLAAC хост не резервирует свой IPv6-адрес и даже не изучает его. Вместо этого он изучает часть адреса (префикс), а затем вычисляет остальную часть собственного IPv6-адреса. А именно: для выбора собственного IPv6-адреса при помощи SLAAC хост использует приведенную ниже последовательность действий.
  1. Выяснение при помощи сообщений NDP RS/RA у любого маршрутизатора префикса IPv6, используемого на канале связи.
  2. Вычисление собственного IPv6-адреса на основании значения идентификатора интерфейса и только что полученного префикса IPv6.
  3. Прежде чем использовать адрес, хост использует процесс DAD, чтобы удостовериться в том, что никакой другой хост не использует тот же адрес.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Компьютеры PC1, PC2 и маршрутизатор R1 объединены в одну сеть VLAN с подсетью IPv6. Хост PC1 хочет послать свой первый пакет IPv6 на хост PC2. Какой протокол или сообщение будет использоваться хостом PC1 для обнаружения MAC-адреса, на который компьютер PC1 должен послать фрейм Ethernet, инкапсулирующий пакет IPv6?

   А) ARP.

   Б) NDP NS.

   В) NDP RS.

   Г) SLAAC.

2. Компьютер PC1 и маршрутизатор R1 подключены к той же сети VLAN и подсети IPv6. Пользователь PC1 вводит команду `ping` для IPv6-адреса хоста, находящегося на дистанционной площадке, так, чтобы пакеты передавались через маршрутизатор R1, стандартный маршрутизатор для компьютера PC1. Параметр стандартного маршрутизатора на компьютере PC1 не задан статически. Какой из следующих ответов указывает протокол или сообщение, используемое компьютером PC1 при попытке узнать IPv6-адрес своего стандартного маршрутизатора?

   А) EUI-64.

   Б) NDP NS.

   В) DAD.

   Г) NDP RS.

3. Какую информацию предоставляет маршрутизатор в анонсе маршрутизатора NDP (RA)? (Выберите два ответа.)

   А) IPv6-адрес маршрутизатора.

   Б) Имя хоста маршрутизатора.

   В) Префикс (префиксы) IPv6 на канале связи.

   Г) IPv6-адрес сервера DHCP.

4. Хост PC1 динамически изучает свои параметры IPv6, используя протокол DHCPv6 с фиксацией состояния. Какой из параметров PC1 должен быть получен от сервера DHCPv6 с фиксацией состояния?

   А) Адрес хоста.

   Б) Длина префикса.

   В) Адрес стандартного маршрутизатора.

   Г) Адрес (адреса) сервера DNS.

5. Хост PC1 динамически изучает свои параметры IPv6, используя автоматическую настройку адреса (SLAAC). Какой из параметров PC1, вероятней всего, будет получен от сервера DHCPv6 без фиксации состояния?

   А) Адрес хоста.

   Б) Длина префикса.

В) Адрес стандартного маршрутизатора.

Г) Адрес (адреса) сервера DNS.

6. Хост PC1 динамически изучает свои параметры IPv6, используя автоматическую настройку адреса (SLAAC). Рассмотрим две части одноадресатного адреса хоста: префикс и идентификатор интерфейса. Какие из перечисленных ниже ответов указывают способ, которым SLAAC изучает или создает значение части идентификатора интерфейса адреса хоста? (Выберите два ответа.)

   А) Предоставляется сервером DHCPv6r.

   Б) Вычисляется хостом с использованием правил EUI-64.

   В) Предоставляется маршрутизатором с использованием сообщения NDP RS/RA.

   Г) Вычисляется хостом с использованием случайного значения.

7. Три маршрутизатора соединены в одну сеть VLAN и подсеть IPv6. Все три маршрутизатора послали сообщения NDP RA в ответ на различные сообщения NDP RS хостов IPv6 с запросами о доступных маршрутизаторах IPv6 в подсети. Сетевой инженер вводит команду `show ipv6 neighbors` на маршрутизаторе R1. Какой из следующих ответов лучше всего описывает вид информации NDP, содержащийся в ее выводе?

   А) Соседи IPv6 (маршрутизаторы и хосты) плюс их MAC-адреса, без отметки маршрутизаторов.

   Б) Соседи IPv6 (маршрутизаторы и хосты), а также их MAC-адреса, с отметкой маршрутизаторов.

   В) Маршрутизаторы IPv6, без информации о не маршрутизаторах, без информации о MAC-адресах.

   Г) Маршрутизаторы IPv6, без информации о не маршрутизаторах, с информацией о MAC-адресах.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 28.3.

Таблица 28.3. Ключевые темы главы 28

Элемент	Описание	Страница
Список	Четыре функции, для которых протокол NDP играет главную роль	801
Список	Описания запросов на получение информации о наличии маршрутизатора NDP и сообщений анонсирования маршрутизатора	802
Рис. 28.2	Пример процесса NDP RS/RA по поиску стандартных маршрутизаторов	802
Список	Описание сообщений запроса и анонса соседнего устройства	804
Рис. 28.4	Пример процесса NDP NS/NA по поиску адресов соседей на канале связи	805
Рис. 28.5	Пример процесса NDP NS/NA по поиску конфликта адресов с соседями на канале связи	806

Окончание табл. 28.3

Элемент	Описание	Страница
Табл. 28.1	Функции протокола NDP	807
Список	Подобия между протоколом DHCP для IPv4 и протоколом DHCP с фиксацией состояния для IPv6	808
Рис. 28.6	Источники параметров IPv6 при использовании протокола DHCP с фиксацией состояния	809
Список	Этапы создания хостом своего IPv6-адреса с использованием SLAAC	812
Рис. 28.9	Создание IPv6-адреса хоста с использованием SLAAC	812
Прим. 28.3	Команда ping6, отданная на компьютере PC1 для хостов R1 и PC2	816

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Ключевые термины

После первого прочтения главы попытайтесь дать определения следующим ключевым терминам. Но не расстраивайтесь, если не все получится сразу, в главе 30 описано, как использовать эти термины на завершающем этапе подготовки к экзамену.

протокол обнаружения соседних устройств (Neighbor Discovery Protocol — NDP), запрос на получение информации о наличии маршрутизатора (Router Solicitation — RS), анонсирование маршрутизатора (Router Advertisement — RA), запрос соседа (Neighbor Solicitation — NS), анонс соседа (Neighbor Advertisement — NA), автоматическая настройка адреса (Stateless Address Autoconfiguration — SLAAC), определение дублирующегося адреса (Duplicate Address Detection — DAD), сервер DHCPv6 с фиксацией состояния (stateful DHCPv6), сервер DHCPv6 без фиксации состояния (stateless DHCPv6), таблица соседних устройств IPv6 (IPv6 neighbor table)

Таблицы команд

Хоть и не обязательно заучивать информацию из таблиц данного раздела, в табл. 28.4 приведен список команд конфигурации, а в табл. 28.5 пользовательские команды главы. Команды стоит запомнить, чтобы лучше понять материал главы и выполнить задания по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 28.4. Команды конфигурации главы 28

Команда	Описание
`ipv6 dhcp relay destination` *адрес_сервера*	Подкоманда интерфейса, включающая агент пересылки DHCP IPv6

Таблица 28.5. Пользовательские команды главы 28

Команда	Описание
`ping {`*`имя_хоста`* `|` *`ipv6-адрес`*`}`	Проверяет маршруты IPv6, посылая пакеты ICMP на хост получателя
`traceroute {`*`имя_хоста`* `|` *`ipv6-адрес`*`}`	Проверяет маршруты IPv6, обнаруживая IP-адреса по маршруту между маршрутизатором и указанным получателем
`show ipv6 neighbors`	Выводит таблицу соседних устройств маршрутизатора IPv6
`show ipv6 routers`	Выводит все соседние маршрутизаторы, анонсирующие себя при помощи сообщений NDP RA

Таблица 28.6. Команды работы с сетевыми хостами главы 28

Команда	Описание
`ipconfig / ifconfig / ifconfig`	Выводит параметры интерфейса, включая адреса IPv4 и IPv6
`ping / ping6 / ping6`	Проверяет маршруты IP, посылая пакет ICMPv6 на хост получателя
`tracert / traceroute6 / traceroute6`	Проверяет маршруты IP, обнаруживая IPv6-адреса по маршруту между маршрутизатором и указанным получателем
`netsh interface ipv6 show neighbors / ndp -an / ip -6 neighbor show`	Выводит таблицу соседних устройств хоста IPv6

Ответы на контрольные вопросы:

1 Б. **2** Г. **3** А и В. **4** В. **5** Г. **6** Б и Г. **7** А.

Глава 29

Реализация маршрутизации по протоколу IPv6

В оставшейся части этой книги, посвященной протоколу IPv6, описывается изучение маршрутизаторами маршрутов IPv6. Самый простой и наиболее распространенный способ изучения маршрутизаторами всех маршрутов заключается в использовании такого динамического протокола маршрутизации IPv6, как открытый протокол поиска первого кратчайшего маршрута (OSPF) версии 3 (OSPFv3). В третьем разделе этой главы обсуждаются основы реализации протокола OSPFv3.

Тем не менее в этой книге пока еще не обсуждались другие упрощенные способы добавления маршрутизаторами маршрутов IPv6 в свои таблицы маршрутизации: подключенные, локальные и статические маршруты. В первом разделе этой главы речь пойдет о том, как протокол IPv6, подобно протоколу IPv4, добавляет подключенные и локальные маршруты на основании IPv6-адреса каждого интерфейса. Во втором разделе рассматривается настройка статических маршрутов IPv6 при помощи ввода команд, в данном случае команды `ipv6 route` вместо команды `ip route` протокола IPv4.

В этой главе рассматриваются следующие экзаменационные темы

IP-адресация (IPv4/IPv6)

Технологические требования для запуска протокола IPv6 совместно с протоколом IPv4 как двойного стека.

Технологии маршрутизации IP

Проверка конфигурации маршрутизатора и сетевого подключения.

Команды Cisco IOS для просмотра базовой информации маршрутизатора и сетевого подключения.

Настройка и проверка конфигурации маршрутизации для статического или стандартного маршрута согласно заданным требованиям маршрутизации.

Различия методов маршрутизации и протоколов маршрутизации:

Статика или динамика.

Пассивные интерфейсы.

Настройка и проверка OSPF (единая область).

Преимущество единой области.

Настройка OSPF v3.

Идентификатор маршрутизатора.

Пассивный интерфейс.

Основные темы

Подключенные и локальные маршруты IPv6

Маршрутизатор Cisco добавляет маршруты IPv6 в свою таблицу маршрутизации IPv6 по нескольким причинам. На настоящий момент большинство читателей вполне могут назвать эти причины, частично потому, что их логика отражает логику использования маршрутизаторов в протоколе IPv4. Маршрутизаторы добавляют маршруты IPv6 на основании следующего.

Методы создания маршрутизаторами маршрутов IPv6

Ключевая тема

- Настройка IPv6-адресов на рабочих интерфейсах (подключенные и локальные маршруты).
- Прямая настройка статического маршрута (статические маршруты).
- Настройка протокола маршрутизации, такого как OSPFv3, на маршрутизаторах, совместно использующих тот же канал связи (динамические маршруты, в данном случае маршруты OSPF).

Правила для подключенных и локальных маршрутов

Маршругизаторы добавляют и удаляют подключенные и локальные маршруты на основании конфигурации и состояния интерфейса. Сначала маршрутизатор ищет все одноадресатные адреса, настроенные на любых интерфейсах, при помощи команды `ipv6 address`. Затем, если интерфейс работает (т.е. команда `show interfaces` отображает для интерфейса "состояние линии up и состояние протокола up"), маршрутизатор добавляет подключенный и локальный маршруты.

ВНИМАНИЕ!

Маршрутизаторы не создают маршруты IPv6 для адресов, локальных в пределах канала связи.

Сами подключенные и локальные маршруты следуют той же общей логике, что и у протокола IPv4. Подключенный маршрут представляет соединенную с интерфейсом подсеть, в то время как локальный маршрут — это маршрут к хосту только для конкретного IPv6-адреса, настроенного на интерфейсе.

В качестве примера рассмотрим маршрутизатор с рабочим интерфейсом, настроенным командой `ipv6 address 2000:1:1:1::1/64`. Маршрутизатор вычислит идентификатор подсети на основании этого адреса и префикса, а полученный маршрут для этой подсети (2000:1:1:1::/64) поместит в таблицу маршрутизации. Маршрутизатор получает также указанный IPv6-адрес и создает маршрут к хосту для этого адреса с длиной префикса /128. (В протоколе IPv4 маршруты хоста имеют длину префикса /32, в то время как в протоколе IPv6 используется длина префикса/128, означающая "только этот адрес".)

Все эти правила для простоты просмотра и изучения приведены ниже.

Ключевая тема

Правила для подключенных и локальных маршрутов IPv6

1. Маршрутизаторы создают маршруты IPv6 на основании всех одноадресатных IPv6-адресов интерфейсов, как указано командой `ipv6 address`, следующим образом:
   - маршрутизатор создает маршрут для подсети (подключенный маршрут);
   - маршрутизатор создает маршрут хоста (с длиной префикса /128) для IPv6-адреса маршрутизатора (локальный маршрут).
2. Маршрутизаторы не создают маршруты на основании адресов, локальных в пределах канала связи, связанных с интерфейсом.
3. Маршрутизаторы удаляют подключенные и локальные маршруты для отказавших интерфейсов и повторно добавляют их, когда интерфейс снова переходит в рабочее состояние (up/up).

Пример подключенных маршрутов IPv6

Хотя концепции подключенного и локального маршрутов IPv6 весьма похожи на таковые у IPv4, лучше понять различие поможет несколько примеров. Для примеров некоторых маршрутов на рис. 29.1 представлена типичная объединенная сеть, используемая в данной главе. На рисунке показаны идентификаторы подсети IPv6. Следующие примеры посвящены подключенным и локальным маршрутам на маршрутизаторе R1.

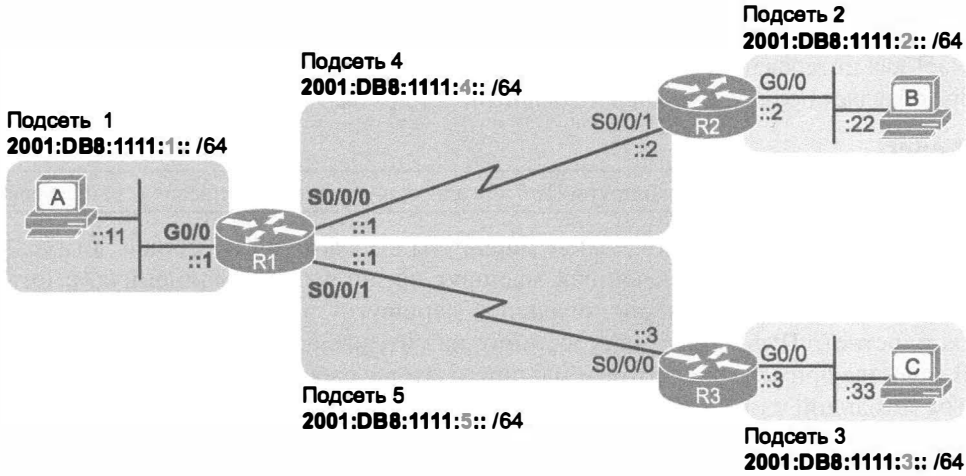


Рис. 29.1. Пример сети, использованной для демонстрации подключенных и локальных маршрутов

На рисунке представлены префиксы IPv6 (подсети) с сокращенной записью IPv6-адресов интерфейса. На рисунке приведена только сокращенная часть идентификатора интерфейса каждого адреса интерфейса. Например, адрес интерфейса G0/0 маршрутизатора R1 начинается со значения 2001:DB8:1111:1 идентификатора подсети, добавленной части ::1, чтобы получилось 2001:DB8:1111:1::1.

Теперь перейдем к примеру подключенных маршрутов. Сначала рассмотрим конфигурацию маршрутизатора R1 на рис. 29.1, приведенную в примере 29.1. Это выдержка из команды `show running-config` на маршрутизаторе R1, демонстрирующая три интерфейса, каждый из которых работает. Обратите также внимание на отсутствие статического маршрута или конфигурации OSPFv3.

Пример 29.1. Настройка адресации IPv6 на маршрутизаторе R1

```
ipv6 unicast-routing
!
interface serial0/0/0
    ipv6 address 2001:db8:1111:4::1/64
!
interface serial0/0/1
    ipv6 address 2001:db8:1111:5::1/64
!
interface gigabitethernet0/0
    ipv6 address 2001:db8:1111:1::1/64
```

На основании рис. 29.1 и примера 29.1 маршрутизатор R1 должен иметь три подключенных маршрута IPv6, как подчеркивается в примере 29.2.

Пример 29.2. Маршруты на маршрутизаторе R1 перед добавлением статического маршрута или маршрута OSPF

```
R1# show ipv6 route
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
   B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
   IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
   ND - Neighbor Discovery, l - LISP
   O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
   ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C     2001:DB8:1111:1::/64 [0/0]
       via GigabitEthernet0/0, directly connected
L     2001:DB8:1111:1::1/128 [0/0]
       via GigabitEthernet0/0, receive
C     2001:DB8:1111:4::/64 [0/0]
       via Serial0/0/0, directly connected
L     2001:DB8:1111:4::1/128 [0/0]
       via Serial0/0/0, receive
C     2001:DB8:1111:5::/64 [0/0]
       via Serial0/0/1, directly connected
L     2001:DB8:1111:5::1/128 [0/0]
       via Serial0/0/1, receive
L     FF00::/8 [0/0]
       via Null0, receive
```

Все три выделенных маршрута приводят те же базовые виды информации, поэтому для обсуждения остановимся на первой паре выделенных строк, представляющей полученный маршрут для подсети 2001:DB8:1111:1::/64. Большая часть информации очевидна: фраза "directly connected" (подключен непосредственно) относится к тому факту, что это подключенный маршрут для идентификатора интерфейса и префикса/длины 2001:DB8:1111:1::/64. Слева расположен кодовый знак "C", указывающий,

что это подключенный маршрут (см. легенду вверху). Обратите также внимание на то, что числа в скобках отражают ту же информацию, что и команда IPv4 `show ip route`: первое число — это административное расстояние, второе — метрика.

Примеры локальных маршрутов IPv6

Продолжая тот же пример, заметим, что на маршрутизаторе R1 должны существовать три локальных маршрута для тех же трех интерфейсов, что и подключенные маршруты. Это действительно имеет место с одним дополнительным локальным маршрутом, но в других целях. Пример 29.3 демонстрирует только локальные маршруты, как указано командой `show ipv6 route local`, с одним выделенным локальным маршрутом для обсуждения.

Пример 29.3. Локальные маршруты IPv6 на маршрутизаторе R1

```
R1# show ipv6 route local
! Легенда пропущена для краткости


L     2001:DB8:1111:1::1/128 [0/0]
       via GigabitEthernet0/0, receive
L     2001:DB8:1111:4::1/128 [0/0]
       via Serial0/0/0, receive
L     2001:DB8:1111:5::1/128 [0/0]
       via Serial0/0/1, receive
L     FF00::/8 [0/0]
       via Null0, receive
```

Рассмотрим несколько фактов для выделенного локального маршрута. Сначала вернемся к конфигурации маршрутизатора R1 в примере 29.1 и обратим внимание на IPv6-адрес интерфейса G0/0 маршрутизатора R1. Этот локальный маршрут демонстрирует точно тот же адрес. Обратите также внимание на длину префикса, /128, означающую, что этот маршрут соответствует пакету, посланному только на этот и только на этот адрес (2001:DB8:1111:1::1).

Статические маршруты IPv6

В то время как маршрутизаторы автоматически добавляют подключенные и локальные маршруты на основании конфигурации интерфейса, статические маршруты требуют непосредственной настройки при помощи команды `ipv6 route`. Проще говоря, некто должен настроить команду, а маршрутизатор помещает подробности команды о маршруте в таблицу маршрутизации IPv6.

Команда `ipv6 route` следует той же общей логике, что и команда IPv4 `ip route` (см. главу 16). Для команды `ip route` протокола IPv4 указывают идентификатор подсети и маску, а для команды `ipv6 route` протокола IPv6 указывают сначала префикс и его длину. Затем соответствующие команды задают направления, куда маршрутизатор должен перенаправить пакеты для подсети получателя или префикса, указав исходящий интерфейс или адрес следующего транзитного маршрутизатора.

На рис. 29.2 представлены концепции (понятия) одиночной команды `ipv6 route`, а также концепции статического маршрута на маршрутизаторе R1 для подсети справа (подсеть 2, или 2001:DB8:1111:2::/64). Статический маршрут на маршрутизаторе R1 для этой подсети начнется с команды `ipv6 route 2001:DB8:1111:2::/64`, сопровождаемой или исходящим интерфейсом (S0/0/0), или IPv6-адресом следующей транзитной точки перехода, или обоими.

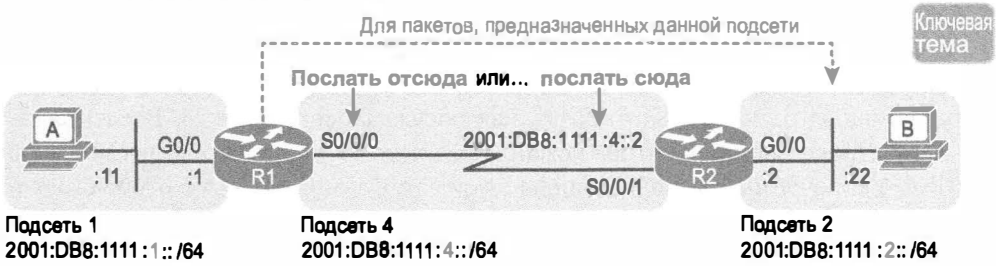


Рис. 29.2. Логика команд статического маршрута IPv6 (маршрут IPv6)

Теперь, когда большинство концепций статической маршрутизации IPv6 понятно, рассмотрим ряд примеров. В частности, примеры настройки статических маршрутов на исходящем интерфейсе, потом с глобальным одноадресатным адресом следующей транзитной точки перехода, а затем с адресом, локальным в пределах канала связи следующей транзитной точки перехода. Завершается данный раздел обсуждением статических стандартных маршрутов IPv6.

Статические маршруты с использованием исходящего интерфейса

Первый пример статического маршрута IPv6 использует параметр исходящего интерфейса. Помните: и для статических маршрутов IPv4, и IPv6, когда команда ссылается на интерфейс, он является локальным интерфейсом. Таким образом, это тот интерфейс на маршрутизаторе, где добавляется команда. В данном случае, как показано на рис. 29.2, команда `ipv6 route` на маршрутизаторе R1 использовала бы интерфейс S0/0/0, как показано вверху примера 29.4.

Пример 29.4. Статические маршруты IPv6 на маршрутизаторе R1

```
! Статический маршрут на маршрутизаторе R1
R1(config)# ipv6 route 2001:db8:1111:2::/64 s0/0/0
```

Пример 29.4 демонстрирует правильный синтаксис маршрута, если при использовании статических маршрутов повсюду в этой объединенной сети необходимо больше статических маршрутов. Например, чтобы обеспечить трафик между хостами A и B, теперь готовится маршрутизатор R1. Хост A перенаправит все свои пакеты IPv6 на свой стандартный маршрутизатор (R1), и маршрутизатор R1 может теперь перенаправлять эти пакеты далее на интерфейс S0/0/0 маршрутизатора R2. Однако маршрутизатор R2 еще не имеет маршрута назад к подсети хоста A, т.е. подсети 1 (2001:DB8:1111:1::/64). Таким образом, полное решение требует большего количества маршрутов.

Пример 29.5 решает эту проблему, предоставляя маршрутизатору R2 статический маршрут для подсети 1 (2001:DB8:1111:1:/64). После добавления этого маршрута хосты A и B должны быть в состоянии обмениваться пакетами друг с другом.

Пример 29.5. Статические маршруты IPv6 на маршрутизаторе R2

```
! Статический маршрут на маршрутизаторе R2
R2(config)# ipv6 route 2001:db8:1111:1::/64 s0/0/1
```

Существует много возможностей для проверки существования статического маршрута и того, могут ли хосты использовать маршрут. Проверить подключение могут команды `ping` и `traceroute`, как обсуждалось в главе 28. Введенная в командной строке маршрутизатора команда `show ipv6 route` отобразит все маршруты IPv6. Укороченный вывод команды `show ipv6 route static` отображает только статические маршруты, как демонстрирует вывод примера 29.6, с пропущенной легендой.

Пример 29.6. Проверка статических маршрутов только на маршрутизаторе R1

```
R1# show ipv6 route static
! Легенда пропущена для краткости
S     2001:DB8:1111:2::/64 [1/0]
       via Serial0/0/0, directly connected
```

Эта команда отображает множество фактов об одном статическом маршруте на маршрутизаторе R1. Сначала код "S" слева идентифицирует маршрут как статический. (Однако последующая фраза "directly connected" (подключенный непосредственно) могла бы ввести в заблуждение, заставив подумать, что это подключенный маршрут; доверяйте коду "S".) Обратите внимание, что префикс (2001:DB8:1111:2::/64) соответствует конфигурации (в примере 29.4) и указывает исходящий интерфейс (S0/0/0).

Команда выводит основную информацию о каждом статическом маршруте, но не сообщает, будет ли использоваться этот маршрут при перенаправлении пакетов определенному получателю. Например, если хост A посылает пакет IPv6 хосту B (2001:DB8:1111:2::22), должен ли маршрутизатор R1 использовать этот статический маршрут? На этот вопрос отвечает команда `show ipv6 route 2001:DB8:1111:2::22`. Эта команда просит, чтобы маршрутизатор указал маршрут, который маршрутизатор использовал бы при перенаправлении пакетов на данный конкретный адрес, как демонстрирует пример 29.7.

Пример 29.7. Отображение маршрута, используемого маршрутизатором R1 для перенаправления пакетов хосту B

```
R1# show ipv6 route 2001:db8:1111:2::22
Routing entry for 2001:DB8:1111:2::/64
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    directly connected via Serial0/0/0
      Last updated 00:01:29 ago
```

Статические маршруты с использованием IPv6-адреса следующей транзитной точки перехода

У статических маршрутов IPv6, ссылающихся на адрес следующей транзитной точки перехода, есть две возможности: одноадресатный адрес соседнего маршрутизатора (глобальный одноадресатный или уникальный локальный) или адрес, локальный в пределах канала связи, для того же соседнего маршрутизатора. Рис. 29.3, модифицированная версия рис. 29.2, обстоятельно объясняет эти две возможности, демонстрируя на сей раз, что маршрутизатор R2 использует глобальный одноадресатный адрес как адрес, локальный в пределах канала связи R2.

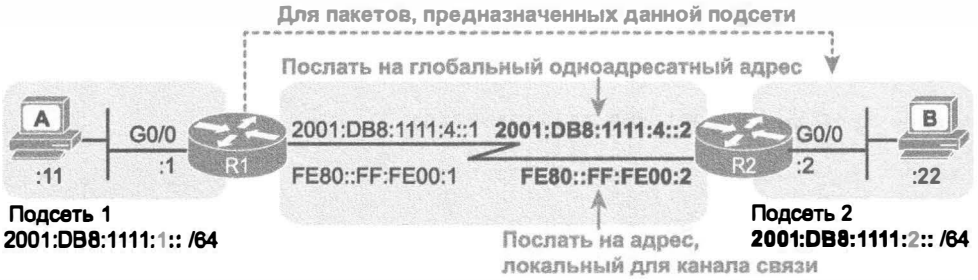


Рис. 29.3. Использование для статических маршрутов одноадресатного адреса или адреса, локального для канала связи, как следующей транзитной точки перехода

Далее представлены примеры: сначала с глобальным одноадресатным адресом, как следующая транзитная точка перехода, а затем с локальным для канала связи.

Пример статического маршрута с глобальным одноадресатным адресом следующей транзитной точки перехода

Этот пример использует объединенную сеть, показанную на рис. 29.3, но с удаленными прежними статическими маршрутами. Таким образом, в начале примера оба маршрутизатора только соединены.

В примере 29.8 маршрутизаторы R1 и R2 добавляют статические маршруты, ссылающиеся на глобальный одноадресатный адрес соседа. Маршрутизатор R1 добавляет маршрут для подсети 2 (справа), в то время как маршрутизатор R2 добавляет маршрут для подсети 1 (слева). Обратите внимание, что пример демонстрирует маршруты в обоих направлениях, чтобы эти два хоста могли посылать пакеты друг другу.

Пример 29.8. Статические маршруты IPv6, использующие глобальные одноадресатные адреса

```
! Первая команда, отданная на маршрутизаторе R1, использует глобальный
! одноадресатный адрес маршрутизатора R2
R1(config)# ipv6 route 2001:db8:1111:2::/64 2001:DB8:1111:4::2
! Следующая команда, отданная на маршрутизаторе R2, использует глобальный
! одноадресатный адрес маршрутизатора R1
R2(config)# ipv6 route 2001:db8:1111:1::/64 2001:db8:1111:4::1
```

Сама команда `ipv6 route` относительно проста. Сосредоточьтесь на маршруте к R1, соответствующем логике, представленной на рис. 29.3. В команде указана подсеть 2 (2001:DB8:1111:2::/64), а затем глобальный одноадресатный адрес маршрутизатора R2 (заканчивающийся на 4::2).

Команды проверки маршрутизатора R1 показаны в примере 29.9. Пример 29.9 демонстрирует две команды; первая выводит только статический маршрут R1 (заданный в примере 29.8). Команда `show ipv6 route 2001:DB8:1111:2::22` в конце примера отображает маршрут R1, используемый при перенаправлении пакетов хосту B, и свидетельствует, что маршрутизатор R1 использует этот новый статический маршрут при перенаправлении пакетов к данному хосту.

Пример 29.9. Проверка статических маршрутов к следующей транзитной точке перехода по глобальному одноадресатному адресу

```
R1# show ipv6 route static
! Легенда пропущена для краткости
S     2001:DB8:1111:2::/64 [1/0]
       via 2001:DB8:1111:4::2

R1# show ipv6 route 2001:db8:1111:2::22/64
Routing entry for 2001:DB8:1111:2::/64
  Known via "static", distance 1, metric 0
  Backup from "ospf 1 [110]"
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:1111:4::2
      Last updated 00:07:43 ago
```

Пример статического маршрута с локальным для канала связи адресом следующей транзитной точки перехода

Статические маршруты, ссылающиеся на локальный в пределах канала связи адрес соседа, отчасти похожи на два предыдущих типа статических маршрутов. Команда `ipv6 route` использует адрес следующей транзитной точки перехода, а именно адрес, локальный в пределах канала связи. Однако команда должна также относиться и к локальному исходящему интерфейсу маршрутизатора. Почему оба адреса? Команда `ipv6 route` не может просто обратиться к локальному для канала связи адресу следующей транзитной точки перехода отдельно, поскольку адрес, локальный в пределах канала связи отдельно, не укажет локальному маршрутизатору, какой исходящий интерфейс использовать.

Интересно, что когда команда `ipv6 route` обращается к глобальному одноадресатному адресу следующей транзитной точки перехода, маршрутизатор может вывести исходящий интерфейс. Приведенный ранее пример 29.8 демонстрирует маршрутизатор R1 со статическим маршрутом IPv6 для IPv6-адреса следующей транзитной точки перехода 2001:DB8:1111:4::2. Маршрутизатор R1 может просмотреть свою таблицу маршрутизации IPv6, увидеть подключенный маршрут, включающий адрес 2001:DB8:1111:4::2, и найти подключенный маршрут от интерфейса S0/0/0 маршрутизатора R1. В результате, имея глобальный одноадресатный адрес следующей транзитной точки перехода, маршрутизатор R1 может вывести правильный исходящий интерфейс (S0/0/0).

Эта логика не годится в случае с локальным для канала связи адресом следующей транзитной точки перехода, так как маршрутизатор не может таким же образом вывести исходящий интерфейс — он должен быть настроен. Пример 29.10 демонстрирует конфигурацию статических маршрутов на маршрутизаторах R1 и R2, упомянутых для этих двух маршрутов, предварительно настроенных в примере 29.8.

Пример 29.10. Статические маршруты IPv6, использующие локальные для канала связи адреса соседей

```
! Первая команда, отданная на маршрутизаторе R1, использует локальный в
! пределах канала связи адрес маршрутизатора R2
R1(config)# ipv6 route 2001:db8:1111:2::/64 S0/0/0 FE80::FF:FE00:2
! ======================================================
! Следующая команда, отданная на маршрутизаторе R2, использует локальный
! в пределах канала связи адрес маршрутизатора R1
R2(config)# ipv6 route 2001:db8:1111:1::/64 S0/0/1 FE80::FF:FE00:1
```

Пример 29.11 демонстрирует проверку конфигурации в примере 29.10, повторяя команды `show ipv6 route static` и `show ipv6 route 2001:DB8:1111:2::22`, используемые в примере 29.9. Обратите внимание, что вывод обеих команд немного отличается в деталях перенаправления. Поскольку новые команды перечисляют и адрес следующей транзитной точки перехода, и исходящего интерфейса, команды `show` указывают также и следующую транзитную точку перехода (адрес, локальный для канала связи) и исходящий интерфейс. Если вернуться к примеру 29.9, то можно увидеть только адрес следующей транзитной точки перехода.

Пример 29.11. Проверка статических маршрутов к следующей транзитной точке перехода по адресу, локальному в пределах канала связи

```
R1# show ipv6 route static
! Легенда пропущена для краткости

S     2001:DB8:1111:2::/64 [1/0]
        via FE80::FF:FE00:2, Serial0/0/0

R1# show ipv6 route 2001:db8:1111:2::22
Routing entry for 2001:DB8:1111:2::/64
  Known via "static", distance 1, metric 0
  Backup from "ospf 1 [110]"
  Route count is 1/1, share count 0
  Routing paths:
    FE80::FF:FE00:2, Serial0/0/0
      Last updated 00:08:10 ago
```

Статические стандартные маршруты

Протокол IPv6 поддерживает концепцию стандартного маршрута, подобную таковой у протокола IPv4. Стандартный маршрут указывает маршрутизатору, что делать с пакетом IPv6, когда он не соответствует никакому другому маршруту IPv6. Логика довольно проста:

- без стандартного маршрута маршрутизатор отбросил бы пакет IPv6;
- со стандартным маршрутом маршрутизатор перенаправит пакет IPv6 по стандартному маршруту.

Стандартные маршруты могут быть особенно полезны в нескольких случаях сетевой топологии. Например, в корпоративной сети, использующей по одному маршрутизатору в каждом филиале и один канал связи WAN в каждой ветви, у маршрутизаторов ветвей есть только один возможный путь для перенаправления пакетов.

В большой сети при использовании протокола маршрутизации маршрутизатор ветви может изучить тысячи маршрутов, причем все они указывают назад к ядру сети по каналу связи WAN.

Вместо протокола маршрутизации маршрутизаторы ветви могли бы использовать стандартные маршруты. Маршрутизатор ветви перенаправил бы весь трафик к ядру сети. Подобный пример с двумя типичными маршрутизаторами ветви справа и основным маршрутизатором площадки слева приведен на рис. 29.4.

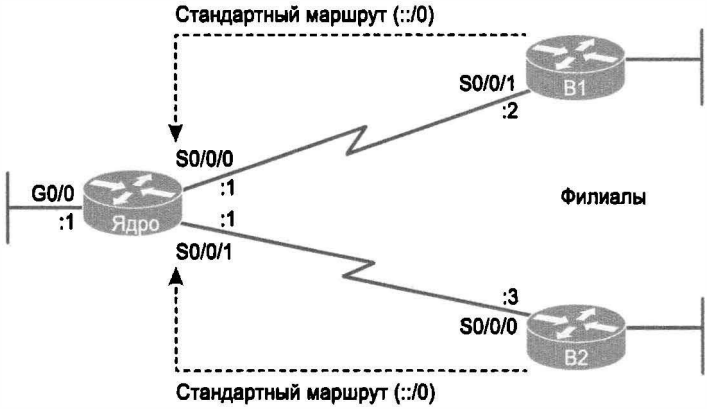


Рис. 29.4. Использование статических стандартных маршрутов в ветвях для перенаправления пакетов назад к ядру

Чтобы задать стандартный маршрут статически, используйте те же правила, которые уже обсуждались в данном разделе, но используйте специфическое значение, чтобы обратить внимание на то, что маршрут стандартный ::/0. Выражаясь буквально, двойное двоеточие (::) — это сокращение IPv6 для всех нулей, а /0 означает длину префикса 0. Эта идея отражает соглашение IPv4 о стандартном маршруте 0.0.0.0/0. В противном случае просто введите команду `ipv6 route`, как обычно.

Пример 29.12 демонстрирует один такой статический стандартный маршрут на маршрутизаторе B1, согласно рис. 29.4. В этом примере используется параметр исходящего интерфейса.

Пример 29.12. Статический стандартный маршрут для маршрутизатора ветви B1

```
! Передача на локальный интерфейс S0/0/1 маршрутизатора B1
B1(config)# ipv6 route ::/0 S0/0/1
```

При протоколе IPv6 маршрутизатор отображает стандартный маршрут более четко, чем при протоколе IPv4. Команда `show ipv6 route` включает маршрут в вывод команды наряду с другими маршрутами. Пример 29.13 демонстрирует стандартный маршрут, отмеченный как "::/0".

Пример 29.13. Статический стандартный маршрут маршрутизатора B1 (использующий глобальный одноадресатный адрес следующей транзитной точки перехода)

```
B1# show ipv6 route static
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
```

```
   B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
   IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
   ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
   O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
   ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S     ::/0 [1/0]
         via Serial0/0/1, directly connected
```

Динамические маршруты и маршруты OSPFv3

Хотя статические маршруты вполне работоспособны, большинство объединенных сетей использует для изучения маршрутов IPv6 ко всем подсетям, не соединенным с локальным маршрутизатором, протокол динамической маршрутизации. В этом последнем разделе главы мы рассмотрим протокол маршрутизации IPv6 (OSPF версии 3), уделяя основное внимание его подобию с протоколом OSPF версии 2.

Данный раздел начинается с рассмотрения некоторых концептуальных и теоретических подробностей о протоколе OSPF версии 3 (OSPFv3). После краткого обсуждения теории мы перейдем к настройке, которая проще, хоть и отличается от таковой у протокола OSPFv2 для IPv4. Раздел завершается несколькими примерами команды `show`, проверяющей правильность работы протокола OSPFv3 и изучения маршрутов IPv6.

Сравнение протоколов OSPF для IPv4 и IPv6

Как и следовало ожидать, протокол OSPFv3 (поддерживающий протокол IPv6) работает подобно протоколу OSPFv2, поддерживающему протокол IPv4. Далее рассматривается часть терминологии и концепций, а также подобия и различия в работе протокола OSPF для IPv6 и для IPv4.

Версии протокола маршрутизации OSPF и протоколы

Когда большинство инженеров упоминают протокол OSPF, они, вероятнее всего, имеют в виду протокол OSPF, используемый с протоколом IPv4, а именно протокол OSPF версии 2 (OSPFv2). Сначала был протокол OSPF версии 1, но затем появилась версия 2 (OSPFv2). Когда в середине 1990-х годов протокол OSPF стал широко использоваться как протокол маршрутизации IPv4, это уже был протокол OSPFv2, а не OSPFv1. Даже в самом начале не было никаких сомнений, использовать ли протокол OSPFv1 или OSPFv2; все использовали протокол OSPFv2, и только его называли OSPF.

Теперь рассмотрим разработку первоначальных протоколов IPv6 в середине недавних 90-х годов. Кроме самого протокола IPv6, следовало модифицировать множество других протоколов, чтобы они поддерживали работу протокола IPv6: ICMP, TCP, UDP и так далее, включая протокол OSPF. Когда рабочая группа модифицировала протокол OSPF для поддержки протокола IPv6, они, конечно, назвали его OSPF версии 3.

Интересно, что протокол OSPFv3 обеспечивает анонсы маршрутов IPv6, но не маршрутов IPv4. Таким образом, протокол OSPFv3 не пытается обеспечить поддержку протокола IPv6 для существовавшего ранее OSPFv2. Хотя протоколы OSPFv2 и OSPFv3 имеют много сходств, считайте их разными протоколами маршрутизации: один для маршрутов IPv4 (OSPFv2), второй для маршрутов IPv6 (OSPFv3).

Поскольку протокол OSPFv3 анонсирует только маршруты протокола IPv6, корпоративная сеть, использующая стратегию двойного стека, фактически должна поддерживать и протокол OSPFv2, и OSPFv3 (если протокол OSPF вообще используется). Действительно ли их базовые концепции очень похожи? Да. Но каждый маршрутизатор выполняет и процесс протокола маршрутизации OSPFv2 и OSPFv3, причем оба процесса формируют соседские отношения, оба отсылают обновления базы данных и оба вычисляют маршрут. Поэтому типичный способ перевода корпоративной сети IPv4 с протокола OSPFv2, подразумевающий поддержку двойного стека IPv4/IPv6 (поддерживающего на каждом хосте/маршрутизаторе и протокол IPv4, и IPv6), использовал бы следующие основные этапы.

Этап 1 До протокола IPv6 компания поддерживала протокол IPv4, использующий протокол OSPFv2

Этап 2 При планировании дополнительной поддержки IPv6 компания планирует использование двойного стека, обеспечивающего поддержку маршрутизации и IPv4, и IPv6 на маршрутизаторах корпоративной сети

Этап 3 Для поддержки маршрутизации IPv6 компания добавляет конфигурацию OSPFv3 ко всем маршрутизаторам, но при сохранении конфигурации OSPFv2, чтобы продолжить поддержку маршрутизации пакетов IPv4

Другие протоколы маршрутизации используют ту же последовательность при переходе на протокол IPv6, хотя и с другими именами. *Протокол маршрутной информации* (Routing Information Protocol — RIP) также имеет две версии, поддерживающие протокол IPv4, с ожидаемыми именами RIP версии 1 (RIPv1) и RIP версии 2 (RIPv2). Для поддержки протокола IPv6 рабочая группа создала новую версию протокола RIP — протокол *RIP следующего поколения* (RIP Next Generation — RIPng), названный так в честь телевизионного сериала "Star Trek". (Именно так.) *Расширенный протокол маршрутизации внутреннего шлюза* (Enhanced Interior Gateway Routing Protocol — EIGRP), собственный протокол маршрутизации Cisco, всегда был известен как просто EIGRP. Но для простоты обсуждения некоторые документы упоминают протокол EIGRP для IPv4 как EIGRP, а протокол EIGRP для IPv6 — как EIGRPv6.

Табл. 29.1 резюмирует терминологию трех основных внутренних протоколов маршрутизации IP.

Таблица 29.1. Терминология внутренних протоколов маршрутизации

	RIP	OSPF	EIGRP
Последняя версия, поддерживающая маршруты IPv4	RIP версии 2 (RIPv2)	OSPF версии 2 (OSPFv2)	EIGRP
Версия, поддерживающая маршруты IPv6	RIP следующего поколения (RIPng)	OSPF версии 3 (OSPFv3)	EIGRP для IPv6 (EIGRPv6)

Сравнение протоколов OSPFv2 и OSPFv3

Концептуально протоколы OSPFv3 и OSPFv2 очень похожи. Например, оба используют логику состояния канала и те же метрики. Этот список довольно длинен,

поскольку у них действительно много сходств. Ниже приведено большинство сходств, обсуждаемых в этой главе и в главе 17.

Сходства протоколов OSPFv2 и OSPFv3

Ключевая тема

- Оба являются протоколами состояния канала.
- Оба используют те же концепции области проекта и термины.
- Оба требуют, чтобы протокол маршрутизации был разрешен на интерфейсе.
- Будучи разрешенными на интерфейсе, оба пытаются обнаружить соседей, подключенных к каналу связи, соединенному с интерфейсом.
- Оба выполняют проверку определенных параметров прежде, чем маршрутизатор станет соседом другим маршрутизаторам (список проверок у протоколов OSPFv2 и OSPFv3 немного разный).
- После того как два маршрутизатора становятся соседями, и протокол OSPFv2, и OSPFv3 обмениваются содержимым своих *баз данных состояния каналов* (Link-State Database — LSDB) и *анонсами состояния каналов* (Link-State Advertisement — LSA), описывающими топологию сети между двумя соседями.
- После обмена всеми анонсами LSA и протокол OSPFv2, и OSPFv3 используют алгоритм *поиска кратчайших маршрутов* (Shortest Path First — SPF) для вычисления наилучшего маршрута к каждой подсети.
- Оба используют ту же концепцию метрики на основании стоимости каждого интерфейса, с теми же стандартными значениями стоимости.
- Оба используют анонсы LSA для описания топологии, но с некоторыми различиями.

Самые большие различия между протоколами OSPFv3 и OSPFv2 заключаются во внутренней организации и конфигурации. У протокола OSPFv3 немного иная структура анонса LSA OSPF; но эти различия не рассматриваются в данной книге. Протокол OSPFv3 использует более непосредственный подход к настройке, разрешая его на каждом интерфейсе с использованием подкоманд интерфейса.

Для последующего сравнения с конфигурацией OSPFv3 на рис. 29.5 показана структура конфигурации для протокола OSPFv2. Он демонстрирует тот факт, что подкоманда `network` протокола OSPFv2 (подкоманда режима конфигурации маршрутизатора) использует IPv4-адрес интерфейса, который затем идентифицирует интерфейс, на котором должен быть разрешен протокол OSPFv2. Другими словами, конфигурация OSPFv2 не упоминает сам интерфейс.

Протокол OSPFv3 непосредственно разрешается на интерфейсе при добавлении в режиме конфигурации интерфейса подкоманды, разрешающей протокол OSPFv3 на этом интерфейсе. Фактически протокол OSPFv3 не использует подкоманду `network` в режиме конфигурации маршрутизатора. Вместо этого он использует подкоманду интерфейса `ipv6 ospf` *идентификатор_процесса* `area` *идентификатор_области*, как показано на рис. 29.6. Эта команда разрешает процесс OSPFv3 на данном интерфейсе и устанавливает область OSPFv3.

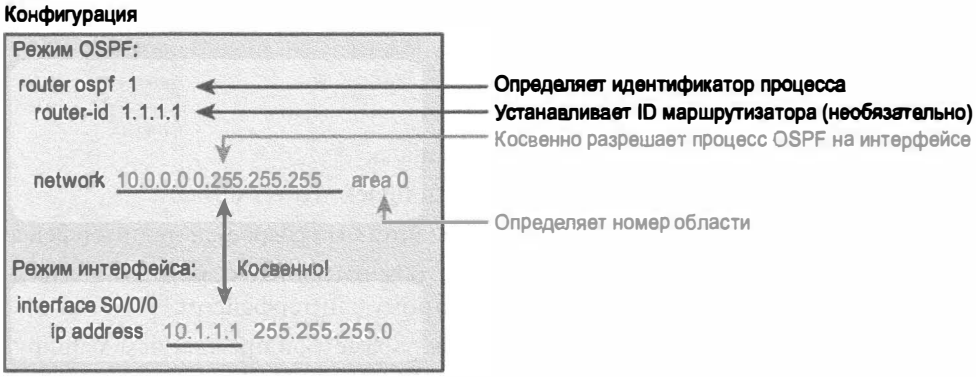


Рис. 29.5. Протокол OSPFv2 косвенно разрешается на интерфейсе

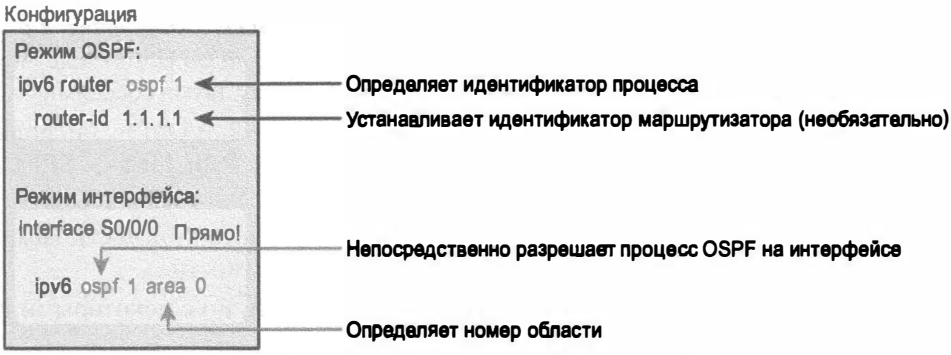


Рис. 29.6. Протокол OSPFv3 непосредственно разрешается на интерфейсе

ВНИМАНИЕ!

Операционная система Cisco IOS фактически поддерживает конфигурацию OSPFv2, использующую тот же стиль команд, представленных для протокола OSPFv3 на рис. 29.6. Операционная система IOS поддерживает только новый, более прямой стиль настройки протокола OSPFv3, как показано на рисунке.

Теперь, когда есть общее представление о подобиях и различиях между протоколами OSPFv3 и OSPFv2, в оставшейся части этого раздела будут приведены примеры настройки и проверки протокола OSPFv3.

Настройка одиночной области OSPFv3

Настройка протокола OSPFv3 осуществляется в несколько простых этапов: выберите и задайте идентификатор процесса, а также разрешите процесс на каждом интерфейсе, назначив правильную область OSPF для каждого интерфейса. Эти детали должны быть указаны в любом плане. Кроме того, в этой книге используется только одиночная область, поэтому все интерфейсы должны быть присвоены той же области.

Одной из потенциальных проблем конфигурации является идентификатор маршрутизатора (RID) протокола OSPFv3.

Протокол OSPFv2 использует 32-разрядный идентификатор RID, выбранный при инициализации процесса OSPF. Поэтому при первой настройке протокола OSPF, или при последующей, или при перезагрузке маршрутизатора процесс OSPFv2 выбирает число, используемое как его RID. Когда процесс OSPFv2 выбирает свой RID, можно ориентироваться на приведенный ниже список.

Правила установки идентификатора маршрутизатора OSPFv3 (RID) Ключевая тема

1. Если подкоманда OSPF `router-id` *rid* настроена, то используется ее значение и игнорируется значение IPv4-адреса интерфейса.
2. Если идентификатор маршрутизатора не установлен командой `router-id`, проверьте все петлевые интерфейсы с настроенным IPv4-адресом и состоянием интерфейса up. Среди них выбирается самый большой в числовом виде IP-адрес.
3. Если ни один из первых двух элементов не предоставляет идентификатор маршрутизатора, он выбирает самый большой в числовом виде IP-адрес из всех других интерфейсов, код состояния интерфейса которых up (первый код состояния). (Другими словами, интерфейс в состоянии up/down будет задействован протоколом OSPF при выборе своего идентификатора маршрутизатора.)

Протокол OSPFv3 также использует 32-разрядный RID и те же правила, что и протокол OSPFv2. Число, как правило, указывается в десятичном представлении с разделительными точками (DDN). Таким образом, идентификаторы маршрутизатора протокола OSPFv3, поддерживающего протокол IPv6, выглядят как IPv4-адреса.

Используя те же правила выбора RID, что и OSPFv2, протокол OSPFv3 оставляет открытой одну прискорбную потенциальную ошибку в конфигурации: маршрутизатор, который не использует команду OSPFv3 `router-id` и не имеет никаких настроенных IPv4-адресов, не может выбрать идентификатор RID. Если у процесса OSPFv3 нет идентификатора RID, процесс не может ни работать правильно, ни установить соседские отношения, ни обмениваться маршрутами.

Эта проблема может быть легко устранена. При настройке протокола OSPFv3, если у маршрутизатора нет IPv4-адресов, удостоверьтесь в настройке RID при помощи подкоманды маршрутизатора `router-id`.

Кроме этой одной небольшой проблемы, настройка протокола OSPFv3 относительно проста. Ниже приведены ее этапы и используемые команды.

Контрольный список настройки протокола OSPFv3 Ключевая тема

Этап 1 Создайте номер процесса OSPFv3 и перейдите в режим конфигурации OSPF для этого процесса, используя глобальную команду `ipv6 router ospf` *идентификатор_процесса*

Этап 2 Удостоверьтесь, что у маршрутизатора есть идентификатор OSPF, используя:

A. Подкоманду маршрутизатора `router-id` *значение_идентификатора*.

B. Предварительно настроенные IPv4-адреса на каждом петлевом интерфейсе, состояние линии которого `up`.

C. Предварительно настроенные IPv4-адреса на каждом рабочем интерфейсе, состояние линии которого `up`

Этап 3 Введите команду `ipv6 ospf` *идентификатор_процесса* `area` *номер_области* на каждом интерфейсе, на котором должен быть разрешен протокол OSPFv3, чтобы разрешить его и установить номер области для интерфейса

Пример конфигурации одиночной области OSPFv3

На рис. 29.7 приведена объединенная сеть, используемая для примера настройки OSPFv3. Здесь только одна область (область 0), а маршрутизаторы R2 и R3 подключены к той же сети VLAN с префиксом IPv6 справа.

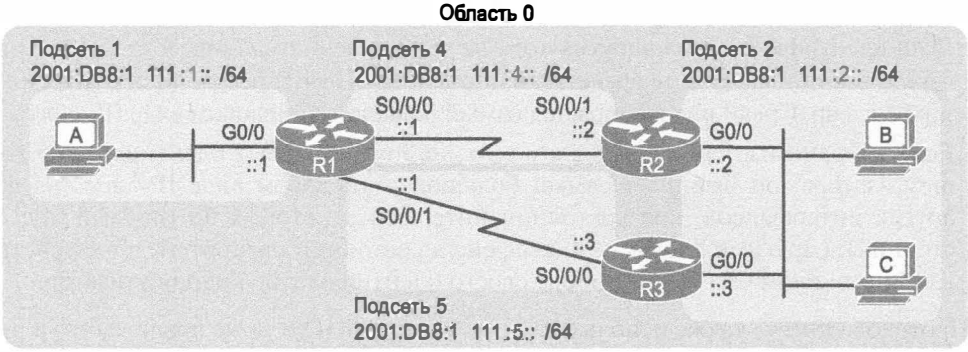


Рис. 29.7. Пример проекта одиночной области в конфигурации OSPFv3

Пример конфигурации OSPFv3 использует следующие требования.

- Все интерфейсы должны быть в области 0, поэтому все команды `ipv6 ospf` *идентификатор_процесса* `area` *номер_области* относятся к области 0.
- Каждый маршрутизатор использует собственный идентификационный номер процесса OSPF, т.е. идентификаторы процесса не должны совпадать у соседних маршрутизаторов OSPFv3.
- При помощи команды `router-id` каждый маршрутизатор непосредственно устанавливает свои идентификаторы маршрутизатора (1.1.1.1, 2.2.2.2 и 3.3.3.3 для маршрутизаторов R1, R2 и R3 соответственно).
- Маршрутизаторы не используют протокол IPv4.

Для начала в примере 29.14 приведен отрывок вывода команды `show running-config` маршрутизатора R1 до добавления конфигурации OSPFv3 командой `ipv6 unicast-routing` и командами `ipv6 address` на каждом интерфейсе.

Пример 29.14. Фрагмент конфигурации IPv6 маршрутизатора R1 (до добавления конфигурации OSPFv3)

```
ipv6 unicast-routing
!
interface serial0/0/0
  no ip address
  ipv6 address 2001:db8:1111:4::1/64
!
interface serial0/0/1
  no ip address
```

```
  ipv6 address 2001:db8:1111:5::1/64
!
interface GigabitEthernet0/0
  no ip address
  ipv6 address 2001:db8:1111:1::1/64
```

Пример 29.15 начинается демонстрацией конфигурации OSPFv3 на маршрутизаторе R1. Обратите внимание, что в настоящий момент на маршрутизаторе R1 не настроен IPv4-адрес, поэтому маршрутизатор R1 не имеет возможности выбрать идентификатор RID протокола OSPFv3; он должен полагаться на конфигурацию команды `router-id`. В примере последовательно происходит следующее.

Этап 1 Инженер добавляет глобальную команду `ipv6 router ospf 1`, создавая процесс OSPFv3

Этап 2 Маршрутизатор безуспешно пытается зарезервировать RID OSPFv3, поэтому выдает сообщение об ошибке

Этап 3 Инженер вводит команду `router-id 1.1.1.1`, чтобы присвоить маршрутизатору R1 идентификатор процесса RID OSPFv3

Этап 4 Инженер вводит команду `ipv6 ospf 1 area 0` для всех трех интерфейсов

Пример 29.15. Дополнительная настройка маршрутизатора R1, разрешающая работу протокола OSPFv3 на трех интерфейсах

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ipv6 router ospf 1
Jan 4 21:03:50.622: %OSPFv3-4-NORTRID: OSPFv3 process 1 could not pick a
router-id, please configure manually
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)#
R1(config-rtr)# interface serial0/0/0
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# interface serial0/0/1
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# interface GigabitEthernet0/0
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# end
R1#
```

В конфигурации на одиночном маршрутизаторе OSPFv3 только два типа параметров могут быть проблемой: идентификатор процесса OSPF и номер области. При проверке конфигурации OSPFv3 на ошибки сначала проверьте номера идентификаторов процесса и удостоверьтесь, что все значения соответствуют данному маршрутизатору. Например, идентификатор процесса команды `ipv6 router ospf` *идентификатор_процесса* должен соответствовать идентификаторам всех подкоманд интерфейса `ipv6 ospf` *идентификатор_процесса* `...`. Другое значение, номер области, должно соответствовать плану на схеме, демонстрирующему, какой области какие интерфейсы должны принадлежать.

В примере 29.16 показана завершенная конфигурация для маршрутизатора R2. В данном случае маршрутизатор R2 использует иной идентификатор процесса

OSPF, чем маршрутизатор R1; идентификаторы процесса на соседних устройствах не должны совпадать с таковыми для OSPFv2 или OSPFv3. Маршрутизатор R2 создает свой процесс OSPFv3 (2), устанавливает его RID (2.2.2.2) и разрешает использование OSPFv3 на всех трех своих интерфейсах при помощи подкоманд интерфейса `ipv6 ospf 2 area 0`.

Пример 29.16. Завершенная конфигурация IPv6, использующая OSPFv3 на маршрутизаторе R2

```
ipv6 unicast-routing
!
ipv6 router ospf 2
router-id 2.2.2.2
!
interface serial0/0/1
ipv6 address 2001:db8:1111:4::2
ipv6 ospf 2 area 0
!
interface GigabitEthernet0/0
ipv6 address 2001:db8:1111:2::2
ipv6 ospf 2 area 0
```

Пассивные интерфейсы OSPFv3

Подобно протоколу OSPFv2, протокол OSPFv3 может быть настроен так, чтобы интерфейсы стали пассивными. У некоторых подсетей IPv6 есть только один маршрутизатор, подключенный к подсети. В таком случае маршрутизатор должен разрешить на интерфейсе протокол OSPFv3, чтобы анонсировать подключенные подсети, но маршрутизатор не должен пытаться обнаруживать соседей OSPFv3 на интерфейсе. Таким образом, инженер может настроить интерфейс как пассивный, указав маршрутизатору сделать следующее.

Ключевая тема

Действия, предпринимаемые и не предпринимаемые для пассивных интерфейсов OSPFv3

- Перестает посылать на интерфейс сообщения OSPF Hello.
- Игнорирует на интерфейсе полученные сообщения Hello.
- Не создает на интерфейсе соседние отношения.
- Продолжает анонсировать все подключенные к интерфейсу подсети.

Интересен тот факт, что конфигурация пассивного интерфейса работает одинаково и у протокола OSPFv2, и у OSPFv3. В примере конфигурации на основании рис. 29.7 только маршрутизатор R1 соединен с подсетью LAN (слева на рисунке), поэтому интерфейс G0/0 маршрутизатора R1 мог бы быть сделан пассивным для OSPFv3. Для этого инженер мог добавить на маршрутизаторе R1 подкоманду `passive-interface gigabitethernet0/0` в режиме конфигурации OSPFv3.

Более подробное обсуждение настройки пассивных интерфейсов для OSPF см. в главе 17.

Проверка состояния протокола OSPFv3 и маршрутов

Для проверки работоспособности протокола OSPFv3 можно использовать два подхода. Можно начать с конца, просмотрев маршруты IPv6, добавленные протоколом OSPFv3. Если в таблицах маршрутизации маршрутизаторов показаны правильные маршруты, то протокол OSPFv3 работает правильно. Но можно следовать в том же порядке, что и протокол OSPF при построении маршрутов: сначала проверьте параметры конфигурации, затем просмотрите соседей OSPF и базу данных OSPF и наконец, маршруты OSPF, добавленные в таблицу маршрутизации.

Когда время имеет значение, просмотрите сначала таблицу маршрутизации. Но в учебных целях имеет смысл повторить этапы от начала до конца, применив множество команд `show` OSPFv3. Остальная часть данного раздела посвящена этим командам OSPFv3 `show` в таком порядке:

- Проверка параметров конфигурации (процесс OSPFv3 и интерфейсы).
- Проверка соседей OSPFv3.
- Проверка базы данных состояния каналов OSPFv3 (LSDB) и LSA.
- Проверка маршрутов OSPFv3.

В данном разделе упоминается широкое разнообразие команд OSPFv3 `show`, имеющих подобные и отличные команды OSPFv2 `show`. Список этих команд `show` приведен в табл. 29.2.

Таблица 29.2. Команды OSPFv3 show и соответствующие им команды OSPFv2

Ключевая тема

Отображает подробности о ...	OSPFv2	OSPFv3
Процесс OSPF	`show ip ospf`	`show ipv6 ospf`
Все источники информации о маршрутизации	`show ip protocols`	`show ipv6 protocols`
Подробности о разрешенных интерфейсах OSPF	`show ip ospf interface`	`show ipv6 ospf interface`
Краткая информация о разрешенных интерфейсах OSPF	`show ip ospf interface brief`	`show ipv6 ospf interface brief`
Список соседей	`show ip ospf neighbor`	`show ipv6 ospf neighbor`
Отчет о LSDB	`show ip ospf database`	`show ipv6 ospf database`
Изученные маршруты OSPF	`show ip route ospf`	`show ipv6 route ospf`

ВНИМАНИЕ!

Все команды OSPFv3 используют те же команды, что и IPv4, но с параметром `ipv6` вместо `ip`.

Проверка параметров конфигурации OSPFv3

Для проверки конфигурации OSPFv3 на маршрутизаторе хорошо работает простая команда `show running-config`. Но в некоторых случаях как реальной жизни, так и многих экзаменационных вопросов нельзя перейти в привилегированный ре-

жим, чтобы использовать такие команды, как `show running-config`. В этих случаях можно воссоздать конфигурацию OSPFv3, используя несколько команд `show`.

Для начала воссоздания конфигурации OSPFv3 просмотрите вывод команды `show ipv6 ospf`. Эта команда выводит информацию о самом процессе OSPFv3. Фактически первая строка вывода, выделенная в примере 29.17, сообщает следующие факты о конфигурации.

- На маршрутизаторе задан идентификатор процесса OSPFv3 1, а значит, была введена команда `ipv6 router ospf 1`.
- Маршрутизатор обладает идентификатором маршрутизатора 1.1.1.1, а значит, была введена либо команда `router-id 1.1.1.1`, либо на некотором интерфейсе маршрутизатора команда `ip address 1.1.1.1` *маска*.

Пример 29.17. Проверка конфигурации процесса OSPFv3

```
R1# show ipv6 ospf
  Routing Process "ospfv3 1" with ID 1.1.1.1
  Event-log enabled, Maximum number of events: 1000, Mode: cyclic
  Initial SPF schedule delay 5000 msecs
  Minimum hold time between two consecutive SPFs 10000 msecs
  Maximum wait time between two consecutive SPFs 10000 msecs
  Minimum LSA interval 5 secs
  Minimum LSA arrival 1000 msecs
  LSA group pacing timer 240 secs
  Interface flood pacing timer 33 msecs
  Retransmission pacing timer 66 msecs
  Number of external LSA 0. Checksum Sum 0x000000
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Graceful restart helper support enabled
  Reference bandwidth unit is 100 mbps
    Area BACKBONE(0)
      Number of interfaces in this area is 3
      SPF algorithm executed 4 times
      Number of LSA 13. Checksum Sum 0x074B38
      Number of DCbitless LSA 0
      Number of indication LSA 0
      Number of DoNotAge LSA 0
      Flood list length 0
```

Выделенные строки в конце примера 29.17 дают некоторое представление об остальной части конфигурации, но с недостаточным количеством подробностей, чтобы отобразить все элементы конфигурации OSPFv3. Выделенные строки свидетельствуют о том, что три интерфейса находятся в этой области и что это опорная область 0. Все эти сообщения находятся ниже строки заголовка вверху части для процесса ospfv3 1. Совместно эти факты свидетельствуют о том, что данный маршрутизатор (R1) использует следующую конфигурацию.

- Подкоманда интерфейса `ipv6 ospf 1 area 0`.
- Маршрутизатор использует эту подкоманду на трех интерфейсах.

Однако команда `show ipv6 ospf` не идентифицирует интерфейсы, на которых запущен протокол OSPFv3. Для поиска этих интерфейсов используется любая из

двух команд в примере 29.18. Рассмотрим сначала команду `show ipv6 ospf interface brief`, выводящую по одной строке для каждого интерфейса, на котором разрешен протокол OSPFv3. Каждая строка указывает интерфейс, идентификатор процесса OSPFv3 (в столбце "PID"), присвоенную интерфейсу область и количество соседей OSPFv3 (в столбце "Nbrs"), знающих этот интерфейс.

Пример 29.18. Проверка интерфейсов OSPFv3

```
R1# show ipv6 ospf interface brief
Interface    PID   Area   Intf ID   Cost   State   Nbrs   F/C
Gi0/0        1     0      3         1      DR      0/0
Se0/0/1      1     0      7         64      P2P    1/1
Se0/0/0      1     0      6         64      P2P    1/1

R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 0):
    GigabitEthernet0/0
    Serial0/0/1
    Serial0/0/0
  Redistribution:
    None
```

ВНИМАНИЕ!

Подобно команде `show ip ospf interface brief`, команда `show ipv6 ospf interface brief` выводит информацию о пассивных и активных интерфейсах OSPFv3.

Вторая половина вывода примера 29.18 — это вывод команды `show ipv6 protocols`, отображающей информацию о каждом источнике маршрутов IPv6 на маршрутизаторе. Эта команда выводит заметно меньше подробностей о протоколе OSPFv3, чем команда `show ip protocol` о протоколе OSPFv2. Тем не менее обе команды отображают интерфейсы, на которых разрешен протокол OSPFv3.

Обе команды в примере 29.18 предоставляют достаточно информации, чтобы засвидетельствовать наличие у маршрутизатора (R1) подкоманды `ipv6 ospf 1 area 0` на трех интерфейсах: G0/0, S0/0/0 и S0/0/1.

Проверка соседей OSPFv3

Проверка соседей OSPFv3 требует беглого взгляда на команду `show ipv6 ospf neighbor`. Для каждого соседа она выводит по одной строке, содержащей ключевые факты об этом соседе. В частности, она указывает RID соседа, а также интерфейс локального маршрутизатора, через который данный сосед установил соседские отношения.

В примере OSPFv3, используемом в данной главе, у каждого маршрутизатора есть два соседа. У маршрутизатора R1 есть два последовательных канала связи, по одному для каждого маршрутизатора R2 и R3. Таким образом, маршрутизатор R1 устанавливает соседские отношения с каждым из этих маршрутизаторов. Оба маршрутизатора, R2 и R3, подключены к той же подсети IPv6 по сети LAN и формируют соседские отношения по этой сети LAN. Ожидаемые соседские отношения OSPFv3 представлены на рис. 29.8.

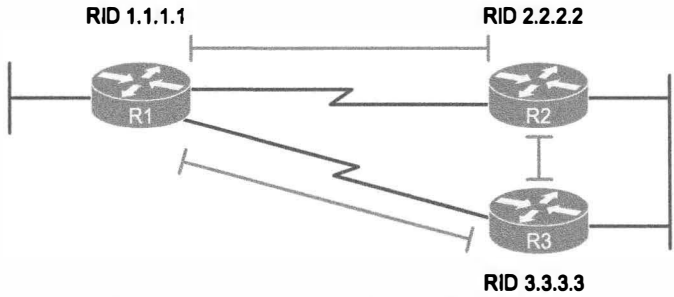


Рис. 29.8. Ожидаемые соседские отношения OSPFv3

Пример 29.19 демонстрирует соседские отношения OSPFv3 на маршрутизаторах R1 и R2. Выделенные части указывают основные сведения: идентификаторы соседних маршрутизаторов (RID), локальный интерфейс и состояние. Состояние "FULL" означает, что соседские отношения работают и что соседи полностью завершили обмен сообщениями LSA.

Пример 29.19. Проверка соседских отношений OSPFv3 на маршрутизаторах R1 и R2

```
! Первая команда, отданная на маршрутизаторе R1, перечисляет R2 и R3
R1# show ipv6 ospf neighbor

Neighbor ID  Pri  State     Dead Time  Interface ID  Interface
3.3.3.3      0    FULL/ -   00:00:39   6             Serial0/0/1
2.2.2.2      0    FULL/ -   00:00:31   7             Serial0/0/0
! ============================================================
! Следующая команда, отданная на маршрутизаторе R2, перечисляет R1 и R3
R2# show ipv6 ospf neighbor

Neighbor ID  Pri  State     Dead Time  Interface  ID Interface
1.1.1.1      0    FULL/ -   00:00:39   6          Serial0/0/1
3.3.3.3      1    FULL/DR   00:00:33   3          GigabitEthernet0/0
```

Прежде чем перейти к следующей теме, уделите еще минуту выводу в примере 29.19. Есть ли здесь что-нибудь, что заставляет думать о протоколе IPv6, а не IPv4? Протокол OSPFv3 использует 32-разрядный RID, указанный в десятичном представлении с разделительными точками. Таким образом, вывод выглядит совершенно так же, как вывод команды `show ip ospf neighbor`. Единственное различие — в самих командах, точнее, в ключевом слове `ipv6`.

Исследование базы данных OSPFv3

Протокол OSPFv3 действительно отличается от протокола OSPFv2 и теорией, и подробностями сообщений LSA. Но в этой книге не рассматриваются подробности сообщений LSA, они коротко обсуждаются во втором томе книги. Чтобы понять различия между протоколами OSPFv2 и OSPFv3, придется намного углубиться в подробности.

Однако простой способ базовой проверки базы LSDB подразумевает проверку *анонсов LSA маршрутизатора типа 1* (Type 1 Router LSA). Оба протокола, OSPFv2 и OSPFv3, используют для каждого маршрутизатора анонс LSA типа 1, описывающий

сам маршрутизатор. У анонса LSA есть идентификатор, равный идентификатору RID этого маршрутизатора. Внутри области база LSDB должна содержать по одному анонсу LSA типа 1 для каждого маршрутизатора в области. Пример 29.20 демонстрирует первый раздел вывода команды `show ipv6 ospf database`, свидетельствуюший о том, изучил ли маршрутизатор анонсы LSA типа 1 от всех маршрутизаторов.

Пример 29.20. Проверка баз LSDB протокола OSPFv3 на маршрутизаторе R1

```
R2# show ipv6 ospf database

    OSPFv3 Router with ID (2.2.2.2) (Process ID 2)

            Router Link States (Area 0)

ADV Router    Age    Seq#         Fragment ID   Link count    Bits
1.1.1.1       452    0x80000002   0             2            None
2.2.2.2       456    0x80000004   0             2            None
3.3.3.3       457    0x80000005   0             2            None

! Строки пропущены для краткости
```

Пример демонстрирует три строки данных ниже строки заголовка в разделе "Router Link States". В этом разделе приведены данные об анонсе LSA маршрутизатора типа 1. В столбце "ADV Router" этого раздела приведены маршрутизаторы, анонсировавшие LSA. В данном случае маршрутизатор R1 (RID 1.1.1.1) знает собственный анонс LSA типа 1 и анонс маршрутизатора R2 (RID 2.2.2.2) и R3 (RID 3.3.3.3).

Исследование маршрутов IPv6, изученных по протоколу OSPFv3

И наконец, реальное доказательство работы OSPFv3, проверка изучения и добавления маршрутизаторами любых маршрутов IPv6 в таблицу маршрутизации IPv6. Этот раздел завершает описание процесса проверки рассмотрением маршрутов IPv6, добавленных протоколом OSPFv3.

При правильной работе маршрутизаторы OSPFv3 изучают достаточно информации, чтобы создать маршруты для всех префиксов IPv6 (подсетей) в объединенной сети. Одним из главных отличий от протокола OSPFv2 является то, что изученные по протоколу OSPFv3 маршруты используют локальный для канала связи адрес следующей транзитной точки перехода. В представленной на рис. 29.9 объединенной сети, проект которой используется для примера конфигурации OSPFv3, маршрутизатор R2 добавляет маршрут к подсети 1 слева (подсеть 2001:DB8:1111:1::/64). На рисунке показано, что маршрутизатор R2 использует локальный в пределах канала связи адрес маршрутизатора R1 как адрес следующей транзитной точки перехода.

Пример 29.21 демонстрирует вывод команды `show ipv6 route ospf` на маршрутизаторе R2 для маршрута, указанного на рис. 29.9. Из особо важного следует отметить следующее.

- Кодовый знак "O" означает "OSPF".
- Цифра 110 в скобках — это административное расстояние OSPF; 65 — метрика для данного маршрута.
- Маршрут выводит как адрес, локальный в пределах канала связи, так и исходящий интерфейс.

Рис. 29.9. Использование адресов, локальных в пределах канала связи

Пример 29.21. Маршруты OSPFv3 на маршрутизаторе R2

```
R2# show ipv6 route ospf
IPv6 Routing Table - default - 9 entries
! Легенда пропущена для краткости

O     2001:DB8:1111:1::/64 [110/65]
        via FE80::FF:FE00:1, Serial0/0/1
O     2001:DB8:1111:5::/64 [110/65]
        via FE80::FF:FE00:3, GigabitEthernet0/0
```

Обзор

Резюме

- Маршрутизаторы добавляют маршруты IPv6 на основании следующего.
  - Настройка IPv6-адресов на рабочих интерфейсах (подключенные и локальные маршруты).
  - Прямая настройка статического маршрута (статические маршруты).
  - Настройка протокола маршрутизации, такого как OSPFv3, на маршрутизаторах, совместно использующих тот же канал связи (динамические маршруты, в данном случае маршруты OSPF).
- Маршрутизаторы добавляют и удаляют подключенные и локальные маршруты на основании конфигурации и состояния интерфейса. Сначала маршрутизатор ищет все одноадресатные адреса, настроенные на любых интерфейсах при помощи команды `ipv6 address`. Затем, если интерфейс работает (т.е. команда `show interfaces` отображает для интерфейса "состояние линии up и состояние протокола up"), маршрутизатор добавляет подключенный и локальный маршруты.
- Для простоты просмотра и изучения эти правила приведены в следующем списке.
  - Маршрутизаторы создают маршруты IPv6 на основании всех одноадресатных IPv6-адресов интерфейсов, как указано командой `ipv6 address`, следующим образом.
  - Маршрутизатор создает маршрут для подсети (подключенный маршрут).
  - Маршрутизатор создает маршрут хоста (с длиной префикса /128) для IPv6-адреса маршрутизатора (локальный маршрут).
  - Маршрутизаторы не создают маршруты на основании адресов, локальных в пределах канала связи, связанных с интерфейсом.
  - Маршрутизаторы удаляют подключенные и локальные маршруты для отказавших интерфейсов и повторно добавляют их, когда интерфейс снова переходят в рабочее состояние (up/up).
- В то время как маршрутизаторы автоматически добавляют подключенные и локальные маршруты на основании конфигурации интерфейса, статические маршруты требуют непосредственной настройки при помощи команды `ipv6 route`. Проще говоря, некто должен настроить команду, а маршрутизатор помещает подробности команды о маршруте в таблицу маршрутизации IPv6.
- Протокол IPv6 поддерживает концепцию стандартного маршрута, подобную таковой у протокола IPv4. Стандартный маршрут указывает маршрутизатору, что делать с пакетом IPv6, когда он не соответствует никакому другому маршруту IPv6. Логика довольно проста:
  - Без стандартного маршрута маршрутизатор отбросил бы пакет IPv6.
  - Со стандартным маршрутом маршрутизатор перенаправит пакет IPv6 по стандартному маршруту.
- Настройка протокола OSPFv3 осуществляется в несколько простых этапов: выберите и задайте идентификатор процесса, а также разрешите процесс на

каждом интерфейсе, назначив правильную область OSPF для каждого интерфейса. Эти подробности должны быть указаны в любом плане. Кроме того, в этой книге используется только одиночная область, поэтому все интерфейсы должны быть присвоены той же области.

- Настройка протокола OSPFv3 относительно проста. Ниже описаны ее этапы и используемые команды.

  Этап 1 Создайте номер процесса OSPFv3 и перейдите в режим конфигурации OSPF для этого процесса, используя глобальную команду `ipv6 router ospf` *`идентификатор_процесса`*

  Этап 2 Удостоверьтесь, что у маршрутизатора есть идентификатор OSPF, используя:

  A. Подкоманду маршрутизатора `router-id` *`значение_идентификатора`*.

  B. Предварительно настроенные IPv4-адреса на каждом петлевом интерфейсе, состояние линии которого up.

  C. Предварительно настроенные IPv4-адреса на каждом рабочем интерфейсе, состояние линии которого up

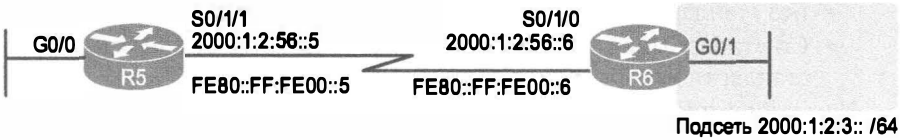
  Этап 3 Введите команду `ipv6 ospf` *`идентификатор_процесса`* `area` *`номер_области`* на каждом интерфейсе, на котором должен быть разрешен протокол OSPFv3, чтобы разрешить его и установить номер области для интерфейса

- Для проверки конфигурации OSPFv3 на маршрутизаторе хорошо работает простая команда `show running-config`. Но в некоторых случаях как реальной жизни, так и многих экзаменационных вопросов нельзя перейти в привилегированный режим, чтобы использовать такие команды, как `show running-config`. В этих случаях можно воссоздать конфигурацию OSPFv3, используя несколько команд `show`.
- Проверка соседей OSPFv3 требует беглого взгляда на команду `show ipv6 ospf neighbor`. Для каждого соседа она выводит по одной строке, содержащей ключевые факты об этом соседе. В частности, она указывает RID соседа, а также интерфейс локального маршрутизатора, через который данный сосед установил соседские отношения.

Контрольные вопросы

Ответьте на эти вопросы. Ответы можно найти на последней странице главы. Полное объяснение ответов приведено в приложении B на веб-сайте.

1. Маршрутизатор был настроен командой `ipv6 address 2000:1:2:3::1/64` на своем интерфейсе G0/1. Маршрутизатор создает также адрес, локальный в пределах канала связи (FE80::FF:FE00:1). Интерфейс работает. Какой из следующих маршрутов маршрутизатор добавит в свою таблицу маршрутизации IPv6? (Выберите два ответа.)



А) Маршрут для 2000:1:2:3::/64.

Б) Маршрут для FE80::FF:FE00:1/64.

В) Маршрут для 2000:1:2:3::1/128.

Г) Маршрут для FE80::FF:FE00:1/128.

2. Инженер должен добавить маршрутизатору R5 статический маршрут IPv6 для префикса 2000:1:2:3::/64, как представлено на рисунке к предыдущему вопросу. В каких из следующих ответов показан правильный статический маршрут IPv6 для этой подсети на маршрутизаторе R5?

А) `ipv6 route 2000:1:2:3::/64 S0/1/1`

Б) `ipv6 route 2000:1:2:3::/64 S0/1/0`

В) `ip route 2000:1:2:3::/64 S0/1/1`

Г) `ip route 2000:1:2:3::/64 S0/1/0`

3. Инженер должен добавить маршрутизатору R5 статический маршрут IPv6 для префикса 2000:1:2:3::/64, как представлено на рисунке к вопросу 1. Какие из следующих ответов демонстрируют правильный статический маршрут IPv6 для этой подсети на маршрутизаторе R5?

А) `ipv6 route 2000:1:2:3::/64 2000:1:2:56::5`

Б) `ipv6 route 2000:1:2:3::/64 2000:1:2:56::6`

В) `ipv6 route 2000:1:2:3::/64 FE80::FF:FE00:5`

Г) `ipv6 route 2000:1:2:3::/64 FE80::FF:FE00:6`

Инженер должен обеспечить поддержку протокола IPv6 на существующем маршрутизаторе R1. На всех интерфейсах маршрутизатора R1 уже функционируют протоколы IPv4 и OSPFv2. Новый проект требует добавить IPv6-адреса и разрешить протокол OSPFv3 на всех интерфейсах. Какие из следующих ответов содержат команды для завершенной конфигурации и обеспечивают поддержку протоколов IPv6 и OSPFv3? (Выберите два ответа.)

А) Глобальная команда `ipv6 router ospf` *`идентификатор_процесса`*.

Б) Команда `router-id` в режиме конфигурации OSPFv3.

В) Команда `network` *`префикс/длина`* в режиме конфигурации OSPFv3.

Г) Команда `ipv6 ospf` *`идентификатор_процесса`* `area` *`идентификатор_области`* на каждом интерфейсе с поддержкой протокола IPv6.

4. Какие из следующих команд отображают интерфейсы, на которых был разрешен протокол OSPFv3? (Выберите два ответа.)

А) `show ipv6 ospf database`

Б) `show ipv6 ospf interface brief`

В) `show ipv6 ospf`

Г) `show ipv6 protocols`

5. Какие из следующих ответов указывают протокол маршрутизации, анонсирующий маршруты IPv6? (Выберите два ответа.)

А) OSPFv6.

Б) OSPFv3.

В) EIGRPv6.

Г) EIGRPv3.

Ключевые темы

Повторите все ключевые темы данной главы, отмеченные пиктограммой "Ключевая тема". Ключевые темы и соответствующие им страницы перечислены в табл. 29.3.

Таблица 29.3. Ключевые темы главы 29

Элемент	Описание	Страница
Список	Методы создания маршрутизаторами маршрутов IPv6	827
Список	Правила для подключенных и локальных маршрутов IPv6	828
Рис. 29.2	Логика команд статического маршрута IPv6 (маршрут IPv6)	831
Список	Сходства протоколов OSPFv2 и OSPFv3	839
Список	Правила установки идентификатора маршрутизатора OSPFv3 (RID)	841
Список	Контрольный список настройки протокола OSPFv3	841
Список	Действия, предпринимаемые и не предпринимаемые для пассивных интерфейсов OSPFv3	844
Табл. 29.2	Команды OSPFv3 show и соответствующие им команды OSPFv2	845

Заполните таблицы и списки по памяти

Распечатайте приложение H (Appendix M) с веб-сайта или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении O (Appendix N) приведены заполненные таблицы и списки для самоконтроля.

Таблицы команд

Хотя и не обязательно заучивать информацию из таблиц данного раздела, в табл. 29.4 приведен список команд конфигурации, а в табл. 29.5 пользовательские команды главы. Команды стоит запомнить, чтобы лучше понять материал главы и выполнить задачи по подготовке к экзамену. Чтобы проверить, насколько хорошо вы запомнили команды, закройте левую сторону таблицы листом бумаги, читайте описания с правой стороны и пытайтесь вспоминать команду.

Таблица 29.4. Команды конфигурации главы 29

Команда	Описание
`ipv6 route` *`префикс/длина адрес_следующего_транзитного_узла`*	Глобальная команда, задающая статический маршрут Ipv6 к Ipv6-адресу следующего транзитного маршрутизатора
`ipv6 route` *`префикс/длина исходящий_интерфейс`*	Глобальная команда, задающая статический маршрут IPv6 для пакетов, покидающих указанный в команде локальный интерфейс маршрутизатора
`ipv6 route` *`префикс/длина адрес_следующего_транзитного_узла исходящий_интерфейс`*	Глобальная команда, задающая статический маршрут IPv6 и для адреса следующей транзитной точки перехода, и для локального исходящего интерфейса маршрутизатора, указанных в команде

Окончание табл. 29.4

Команда	Описание
`ipv6 route ::/0 {[`*адрес_следующего_транзитного_узла*`] [`*исходящий_интерфейс*`]}`	Глобальная команда, задающая стандартный статический маршрут IPv6, с подробностями перенаправления (исходящий интерфейс, адрес следующей транзитной точки перехода), работающая так же, как и версии команды `ipv6 route` не для стандартных маршрутов
`ipv6 router ospf` *идентификатор_процесса*	Переводит в режим конфигурации OSPFv3 для указанного процесса
`router-id` *идентификатор*	Подкоманда OSPF, статически устанавливающая идентификатор маршрутизатора
`ipv6 ospf` *идентификатор_процесса* `area` *номер_области*	Подкоманда интерфейса, включающая протокол OSPFv3 на интерфейсе для заданного процесса и определяющая область OSPFv3
`passive-interface` *тип номер*	Подкоманда OSPF, делающая протокол OSPF пассивным на заданном интерфейсе или субинтерфейсе
`passive-interface default`	Подкоманда OSPF, изменяющая стандартное состояние протокола OSPF для интерфейсов на пассивное вместо активного
`No passive-interface` *тип номер*	Подкоманда OSPF, делающая протокол OSPF активным (не пассивный) на заданном интерфейсе или субинтерфейсе

Таблица 29.5. Пользовательские команды главы 29

Команда	Описание
`show ipv6 route [ospf]`	Выводит список маршрутов в таблице маршрутизации, изученных по протоколу OSPFv3
`show ipv6 ospf`	Выводит параметры протокола маршрутизации и текущие значения таймеров для протокола OSPFv3, а также идентификатор маршрутизатора OSPFv3
`show ipv6 ospf interface brief`	Выводит по одной строке для каждого интерфейса с выполняющимся протоколом OSPFv3 с указанием базовых параметров, таких как процесс OSPFv3, номер области и цена интерфейса
`show ipv6 ospf neighbor [`*rid_соседа*`]`	Выводит список соседей и их текущее состояние по интерфейсам, а также (дополнительно) подробности о маршрутизаторе, идентификатор которого указан в команде
`show ipv6 ospf database`	Выводит отчет по анонсам LSA в базе LSDB локального маршрутизатора (по одной строке на каждый анонс LSA)
`show ipv6 protocols`	Выводит краткую информацию, как и команда `show ip protocols` протокола IPv4, включая все средства, доступные маршрутизатору для изучения или создания маршрутов IPv6, а также интерфейсов, на которых включен протокол маршрутизации

Ответы на контрольные вопросы:

1 А и В. **2** А. **3** Б. **4** А и Г. **5** Б и Г. **6** Б и В.

Обзор части VII

Проследите свой прогресс изучения материала части по контрольному списку в следующей таблице. Подробно задачи описаны ниже.

Контрольный список обзора части VII

Задача	Первая дата завершения	Вторая дата завершения
Повторите вопросы из обзоров глав		
Ответьте на вопросы обзора части		
Повторите ключевые темы		
Создайте диаграмму связей IPv6-адресации		
Создайте диаграмму связей команд конфигурации и проверки		

Повторите вопросы из обзора главы

Ответьте снова на вопросы обзоров глав этой части книги, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров глав только этой части приведена в разделе "Как просмотреть вопросы только обзоров глав конкретной части" введения к книге.

Ответы на вопросы

Ответьте на вопросы обзора этой части книги, используя программное обеспечение PCPT. Инструкция по запуску программного обеспечения PCPT с вопросами обзоров только этой части приведена в разделе "Как просмотреть вопросы только обзоров частей" введения к книге.

Ключевые темы

Снова просмотрите темы прочитанных глав, отмеченные пиктограммой "Ключевая тема". Если понятны не все их подробности, уделите время повторному изучению.

Создайте диаграмму связей IPv6-адресации

Адресация — самое большое различие между протоколами IPv4 и IPv6. Обдумайте IPv6-адреса с нескольких сторон: как термин, как структуру, как тип и как нечто, связанное с адресацией. Затем создайте диаграмму связей, резюмирующую все концепции адресации и термины в одной диаграмме связей.

Размышляя об адресации, попытайтесь организовывать информацию так, как вам нравится. Единственно правильной организации здесь нет. Но если хотите получить некий ориентир по категоризации информации, то некоторые из концепций и терминов можно организовать по типам адресов. Например, одним типом могли бы быть адреса, локальные в пределах канала связи. В этой части диаграммы связей

можно указать все термины и факты об адресах, локальных в пределах канала связи, как показано на рис. Ч7.1.

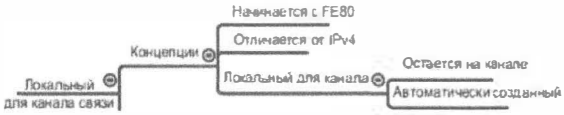


Рис. Ч7.1. Пример диаграммы связей для ветви локального канала связи

Большинство концепций адресации IPv6 в этой книге приведено в главах 25–27. Попытайтесь соотнести в диаграмме все термины адресации из разделов ключевых терминов этих глав наряду со всеми концепциями и значениями адресации IPv6, которые идентифицируют адрес или являются специфическим типом IPv6-адреса.

Создайте диаграмму связей команд конфигурации и проверки

Не обращаясь к своим заметкам или тексту книги, создайте диаграмму связей всех команд маршрутизатора IPv6, которые сможете вспомнить. Задача этого упражнения заключается в том, чтобы помочь запомнить команды. Оно не сосредоточивается на деталях и каждом отдельном параметре каждой команды или даже их значении. Цель в том, чтобы помочь организовать команды в памяти и вспомнить их, когда они встретятся в реальности или на экзамене.

Организуйте свои команды сначала по главной теме. Затем разделите команды для выбранной темы на основании команды настройки или проверки. Вот некоторые из возможных основных темы:

адреса, статические маршруты, OSPFv3

Ответы приведены в приложении П (Appendix O) на веб-сайте, но ваши диаграммы связей могут выглядеть иначе.

Диаграммы связей обзора части VII

Диаграмма	Описание	Где сохранен результат
1	Диаграмма связей IPv6-адресации	
2	Диаграмма связей команд IPv6	

Часть VIII. Подготовка к экзамену

Глава 30. "Подготовка к сертификационному экзамену"

Подготовка к сертификационному экзамену

Поздравляем! Вы сделали это. Книга прочитана, и теперь пришло время заканчивать подготовку к экзамену. Эта глава поможет подготовиться и сдавать экзамен двумя способами.

Глава начинается с описания самого экзамена. Содержимое и темы уже известны. Теперь необходимо подумать, что будет во время экзамена и что необходимо сделать за несколько недель перед сдачей экзамена. На настоящий момент все, что вы делаете, должно быть сосредоточено на подготовке к сдаче, чтобы можно было закончить наконец это грандиозное предприятие.

Второй раздел этой главы дает некоторое представление об экзаменационных задачах и окончательной подготовке к экзаменам ICND1, ICND2 или CCNA.

Советы о самом экзамене

Теперь, изучив основной материал этой книги, можете зарегистрироваться для сдачи экзамена Cisco ICND1, ICND2 или CCNA, прийти и сдать экзамен. Но если уделить немного времени обдумыванию самого события экзамена, узнать чуть больше о пользовательском интерфейсе реальных экзаменов Cisco и окружении в сертификационных центрах, то подготовка окажется гораздо лучше, особенно если это первый экзамен Cisco. В этом первом из трех главных разделов главы дано несколько советов об экзаменах Cisco и самом событии экзамена.

Изучите типы вопросов используя руководство по сертификационным экзаменам Cisco

За оставшиеся до экзамена недели стоит подумать о различных типах экзаменационных вопросов и выработать план ответа на них. Один из наилучших способов изучения экзаменационных вопросов — это воспользоваться руководством по сертификационным экзаменам Cisco (Cisco Exam Tutorial).

Руководство доступно на сайте `www.cisco.com` (достаточно поискать "exam tutorial"). На веб-странице руководства находится также сделанное во Flash представление пользовательского интерфейса экзамена, позволяющее попрактиковаться в сдаче экзамена. В пользовательском интерфейсе экзамена стоит опробовать следующее.

- Пробуйте щелкнуть на кнопке Next (Далее) на вопросе с многовариантным выбором одного ответа и убедитесь, что экзаменационное программное обеспечение укажет на слишком большое количество ответов.
- В вопросе с многовариантным выбором нескольких ответов выберите минимум ответов и, щелкнув на кнопке Next (Далее), посмотрите ответ пользовательского интерфейса.

- Попробуйте в вопросе с перетаскиванием перетащить ответы в некую область, а затем перетащите его назад в первоначальную область. (Это может произойти на реальном экзамене, если вы передумали при ответе на вопрос.)
- В вопросе с имитацией (Simulation question) сначала удостоверьтесь, что можете получить доступ к интерфейсу командной строки (CLI) на одном из маршрутизаторов. Для этого следует шелкнуть на пиктограмме компьютера, подключенного к консоли маршрутизатора; консольный кабель представлен пунктирной линией, а сетевые кабели показаны сплошными линиями.
- Все еще в вопросе с имитацией удостоверьтесь, что смотрите на область прокрутки вверху, а не сбоку, и видите окно эмулятора терминала.
- В вопросе с имитацией удостоверьтесь, что можете переключаться между окном топологии и окном эмулятора терминала шелчком на кнопке Show topology (Показать топологию) и Hide topology (Скрыть топологию).
- В тестлете (Testlet question) ответьте на один вопрос с многовариантным выбором, перейдите ко второму и ответьте, а затем вернитесь к первому вопросу, подтвердив, что в тестлете можно перемещаться между вопросами.
- Снова в тестлете щелкните на кнопке Next (Далее), чтобы увидеть всплывающее окно, используемое Cisco для подтверждения перехода далее. Тестлеты фактически позволяют дать меньше ответов и все равно перейти далее. После щелчка для выхода из тестлета уже нельзя вернуться, чтобы изменить ответ на любой из этих вопросов.

Рассчитывайте свое время относительно количества вопросов

В день экзамена необходимо следить за временем. Продвигаясь слишком медленно, можно упустить время и не успеть ответить на все вопросы. Слишком быстрое продвижение тоже вредно: быстро отвечая на вопросы, вы можете не понять вопрос полностью или допустить ошибку. Поэтому необходимо быть в состоянии так или иначе знать, достаточно ли быстро идет процесс, чтобы ответить на все вопросы, не торопясь.

Пользовательский интерфейс экзамена демонстрирует немало полезной информации, в том числе таймер обратного отсчета, а также счетчик вопросов. Счетчик вопроса демонстрирует количество вопросов, на которые даны ответы, а также все количество вопросов на экзамене.

К сожалению, все вопросы нельзя считать равными, и это не дает точной оценки времени. Например, если экзамен занимает 90 минут и насчитывает 45 вопросов, то на каждый вопрос приходится по две минуты. Если ответ на 20 вопросов занял 40 минут, вполне можно уложиться в срок. Однако такую оценку затрудняет несколько факторов.

В первую очередь, корпорация Cisco не указывает заранее точного количества вопросов для каждого экзамена. Например, на веб-сайте Cisco экзамен CCNA мог бы быть указан как насчитывающий от 45 до 55 вопросов. (У экзаменов ICND1 и ICND2 подобные диапазоны.) Но вы не узнаете, сколько вопросов будет на вашем экзамене, пока он не начнется, точнее, пока вы не щелкнете на кнопке Start exam (Начать экзамен).

Кроме того, некоторые вопросы (их называют *прожигателями времени* (time burners)) явно требуют много больше времени на ответ.

- **Вопросы нормального времени**. Вопросы с многовариантным выбором и перетаскиванием, приблизительно по 1 минуте на каждый.
- **Прожигатели времени**. Симлеты и тестлеты, примерно по 6–8 минут на каждый.

И наконец, при наличии 45–55 вопросов на одном экзамене тестлеты и симлеты могут содержать по несколько вопросов с многовариантным выбором, а в счетчике вопросов они считаются как один вопрос. Например, если тестлет содержит четыре встроенных вопроса с многовариантным выбором, то счетчик вопросов покажет его как один вопрос.

ВНИМАНИЕ!

Хотя компания Cisco никак не объясняет, почему одним может достаться 45 вопросов, а другим 55 на том же экзамене, это, возможно, связано с тем, что в наборе из 45 вопросов могло бы быть больше прожигателей времени, делающих эти два набора эквивалентными.

Необходим план распределения времени, не отвлекающий от экзамена. Он мог бы включать примерно следующие предположения.

50 вопросов, 90 минут — это немного меньше, чем две минуты на вопрос, и на основании этого можно предположить, сколько вопросов прожигателей времени еще не встретилось.

Независимо от того, как планировалось распределить время, подумайте об этом перед днем экзамена. Вполне можно использовать метод, приведенный в следующем разделе.

Предлагаемый метод распределения времени

Для распределения времени с учетом вероятного наличия прожигателей времени применима следующая математика. Вы не обязаны использовать этот метод, но он прост и использует только целые числа. По мнению автора, он дает достаточно близкую оценку времени.

Концепция проста. Вот простое вычисление, позволяющие оценить использованное до сих пор время:

количество пройденных вопросов + 7 минут на каждого прожигателя времени.

Потраченное время покажет таймер, а на основании его значения можно выяснить следующее:

- Использовано именно столько времени или немного больше: хронометраж в порядке.
- Использовано меньше времени: вы опережаете график.
- Использовано заметно больше времени: вы отстаете от графика.

Например, если уже закончено 17 вопросов, 2 из которых были прожигателями, оценка времени составит 17 + 7 + 7 = 31 минуту. Если ваше фактическое время со-

ставляет 31 минуту или, возможно, 32-33 минуты, вы укладываетесь в график. Если времени потрачено меньше 31 минуты, вы опережаете график.

Математика довольно проста: пройденные вопросы плюс 7 минут на каждого прожигателя и ощущение того, укладываетесь ли вы в срок.

ВНИМАНИЕ!

Эта математика приблизительна; автор не дает никаких гарантий, что этот подход даст точный прогноз на каждом экзамене.

Другие рекомендации

Вот еще несколько рекомендаций по подготовке перед экзаменом.

- Купите беруши. В некоторых сертификационных центрах они есть, но чтобы не рисковать, лучше быть подготовленным. Сертификационные центры, как правило, располагаются в помещениях неких компаний, где люди работают, или в учебных центрах, поэтому в соседних помещениях слышны разговоры и другой шум. (Наушники, как электронные устройства, запрещены.)
- Некоторым нравится в первые минуты экзамена сделать несколько заметок для справки. Например, записать таблицу магических чисел для поиска идентификаторов подсети IPv4. Если вы планируете делать это, попрактикуйтесь в составлении таких заметок. Перед каждым тренировочным экзаменом практикуйтесь в создании таких заметок, точно так же, как собираетесь сделать это на реальном экзамене.
- Спланируйте свой визит в сертификационный центр с достаточным запасом времени, чтобы не спешить и сделать все вовремя.
- Если вы нервничаете перед экзаменами, применяйте свои любимые методики расслабления за несколько минут перед каждым тренировочным экзаменом, чтобы быть готовым использовать их.

Советы на день экзамена

Я надеюсь, экзамен пройдет успешно. Конечно, чем лучше подготовка, тем выше шансы преуспеть на экзамене. Но и эти небольшие советы помогут сосредоточить все усилия в день экзамена.

- Перед экзаменом ляжте спать пораньше, не учите допоздна. Ясность мысли важней, чем один дополнительный факт, особенно потому, что экзамен требует больше анализа и размышлений, а не просто запоминания фактов.
- Если вы не принесли беруши, спросите их в сертификационном центре, даже если не предполагаете использовать их. Никогда не знаешь, что может пригодиться.
- Вы можете принести личные вещи в здание сертификационного центра, но не в то помещение, где проходит сдача экзамена. Поэтому не берите с собой по возможности ничего лишнего. Если есть безопасное место, чтобы оставить портфель, кошелек, электронику и т.д., оставьте их там. (В сертификационном центре должно быть место для хранения вещей.) Проще говоря, чем

меньше принесете, тем меньше придется волноваться о сохранности. (Автора, например, попросили снять даже механические наручные часы, причем не раз.)

- Сертификационный центр выдаст ламинированный лист и карандаш, чтобы делать заметки. (Персонал центра, как правило, не позволяет приносить бумагу и ручки в помещение, даже если они взяты в самом центре.)
- Отправляйтесь в сертификационный центр заранее, чтобы не торопиться.
- Планируйте сходить в уборную перед визитом в сертификационный центр. Если не получится, то можно воспользоваться туалетом в сертификационном центре, персонал, конечно, поможет его найти и предоставит время перед началом экзамена.
- Не пейте литр сока перед визитом в сертификационный центр. После начала экзамена таймер не будет останавливаться, пока вы бегаете в туалет.
- В день экзамена используйте любые методики расслабления, которые практикуете, чтобы сосредоточить свой ум, пока ожидаете сдачи экзамена.

Обзор экзамена

Этот обзор завершает материалы плана изучения, предложенного в данной книге. На настоящий момент уже прочитаны все главы и выполнены упражнения обзоров глав и частей. Теперь необходимо выполнить завершающие упражнения и действия, прежде чем переходить к сдаче экзамена, как описано в данном разделе.

В данном разделе предлагается несколько новых действий, а также повторяется несколько старых. Но и новые и старые действия сосредоточены на заполнении пробелов в знаниях и закреплении навыков, что завершает процесс изучения. Повторение некоторых задач из обзоров глав и частей поможет подготовиться к сдаче экзамена; таким образом, данный раздел требует уделить время ответам на экзаменационные вопросы.

Здесь рекомендуется несколько типов заданий и предоставляется несколько таблиц для отслеживания каждого действия. Ниже приведены основные категории.

- Практика на скорость.
- Пробная сдача экзаменов.
- Поиск пробелов в знаниях (слабых мест).
- Настройка и проверка функций из CLI.
- Повторение задач из обзоров глав и частей.

Практика создания подсетей и другие навыки, связанные с математикой

Нравится вам это или нет, но некоторые вопросы на экзаменах Cisco ICND1, ICND2 и CCNA требуют выполнения некоторых математических действий. Для сдачи следует знать математику. Следует также знать, какой процесс и когда использовать.

Экзамены Cisco сдаются на время. Необходимость в математических вычислениях возникает неожиданно и достаточно часто, поэтому, если вы вычисляете медленно или должны записывать все подробности вычислений, сейчас хорошее время решить эти вопросы. (Двумя главными причинами, по которым люди не успевали закончить экзамен вовремя, как слышал автор, являются низкая скорость вычислений и низкая скорость выполнения заданий на эмуляторе CLI.)

Однако смотрите на эти математические процессы и недостаток времени позитивно, а не негативно. Об этой проблеме вам известно сейчас, перед экзаменом. Вы знаете, что если продолжите практиковаться в создании подсетей и другой математике, то станете быстрее и лучше. С приближением дня экзамена, если есть пара свободных минут, старайтесь побольше практиковаться в чем-нибудь, например в создании подсетей. Смотрите на это как на способ подготовки, чтобы не волноваться о нехватке времени в день экзамена.

В табл. 30.1 приведен список тем этой главы, требующих как математических вычислений, так и скорости. В табл. 30.2 приведен список элементов, для которых важнее математические вычисления или процессы, чем скорость. На этом этапе обучения следует уже уверенно находить правильные ответы на эти виды задач. Теперь самое время закрепить свои навыки в получении правильных ответов, чтобы сократить нехватку времени на экзаменах.

ВНИМАНИЕ!

Время на решение задач в таблице выбрано автором, чтобы дать общее представление. Если некоторые задачи выполняются немного медленнее, это вовсе не означает, что на экзамене будет провал. Но если почти каждая задача занимаете в несколько раз больше времени, то проблемы со временем будут.

Таблица 30.1. Математические действия, требующие скорости вычисления

Глава	Действие	Отличная скорость (секунды)	Дата/время самопроверки	Дата/время самопроверки
12	Найти ключевые факты о классовой сети по одноадресатному IPv4-адресу	10		
13	Преобразовать любую маску в одном формате в два других формата маски	10		
13	По IPv4-адресу и маске найти номер сети, биты подсети и хоста, а также количество подсетей и хостов в подсети	15		
14	По IPv4-адресу и маске найти резидентскую подсеть, широковещательный адрес подсети и диапазон пригодных для использования адресов	20–30		
19	По набору требований к маске выбрать наилучшую маску подсети	15		
19	По классовой сети и одной маске найти все идентификаторы подсети	45		

Таблица 30.2. Математические действия, для которых скорость вычислений не столь важна

Глава	Действие	Дата/время самопроверки	Дата/время самопроверки
20	Найти перекрытия VLSM, затрагивающие 5-6 подсетей		
20	Добавить подсети VLSM, затрагивающие 5-6 подсетей		
21	Найти наилучший суммарный маршрут, затрагивающий четыре маршрута		
22	Создать команду ACL, соответствующую адресу подсети		
22	Перечислить адреса, соответствующие одной команде ACL		
25	Найти наилучшее сокращение одного IPv6-адреса		
27	Найти IPv6-адреса одного интерфейса маршрутизатора в формате EUI-64		

Практические задачи на математические действия, перечисленные в табл. 30.1 и 30.2, приведены в конце соответствующих глав. Например, для многих вопросов создания подсетей можно составить собственные задачи и проверить свою работу с любым калькулятором подсетей. Кроме того, для всех этих глав есть соответствующие приложения на веб-странице книги с дополнительными практическими задачами. И наконец, дополнительные практические задачи есть на блогах автора.

Пробная сдача экзаменов

Однажды придется сдавать реальный экзамен Cisco в сертификационном центре. Поэтому пришло время попрактиковаться в пробной сдаче с максимально возможным подобием.

Пробный экзамен использует программное обеспечение Pearson IT Certification Practice Test (PCPT), позволяющее опробовать большинство тех же задач реального экзамена Cisco. Это программное обеспечение задает вопросы, представляя таймер обратного отсчета в окне. После ответа на вопрос к нему больше нельзя вернуться (как и на экзаменах Cisco). Если время закончилось, вопросы без ответа будут считаться неправильными.

Процесс сдачи пробного экзамена на время поможет подготовиться в трех ключевых направлениях.

- Само событие пробного экзамена, включая нехватку времени, научит внимательно читать и запоминать прочитанное на протяжении достаточно длительного времени.
- Позволит выработать критически важные интеллектуальные навыки анализа и исследования сетевого сценария.
- Обнаружить пробелы в знаниях сети, чтобы можно было изучить эти темы перед реальным экзаменом.

В максимально возможной степени отработайте событие пробного экзамена, как будто сдаете реальный экзамен Cisco в сертификационном центре Vue. Ниже приведено несколько советов, позволяющих сделать пробную сдачу экзамена более значащей, а не просто еще одним рутинным событием перед днем экзамена.

- Уделите пару часов на сдачу 90-минутного пробного экзамена с ограничением по времени.
- Создайте список того, что предполагаете сделать на протяжении десяти минут перед реальным событием экзамена. Затем представьте себя выполняющим эти действия. Перед сдачей каждого пробного экзамена попрактикуйтесь в этих действиях. (В предыдущем разделе "Советы на день экзамена" содержатся рекомендации о том, что делать в эти предшествующие десять минут.)
- В экзаменационное помещение ничего принести нельзя, поэтому уберите все заметки и вспомогательные материалы с рабочего места перед пробной сдачей экзамена. Использовать можно только чистый лист, ручку и свои знания. Не используйте на своем компьютере ни калькулятор, ни блокнот, ни веб-браузеры, ни любые другие приложения.
- В реальной жизни вам могут мешать, но если это возможно, попросите окружающих оставить вас в покое на время сдачи пробного экзамена. Если все же придется сдавать пробный экзамен в шумной обстановке, наушники или беруши позволят снизить раздражающее воздействие.
- Не полагайтесь на удачу. Отвечайте только тогда, когда уверены. Если вопрос непонятен, к нему можно вернуться и обдумать впоследствии.

Пробная сдача экзамена ICND1

Поскольку вы читаете эту главу в первом томе, то, вероятнее всего, готовитесь либо к экзамену ICND1, либо CCNA. Экзаменационное программное обеспечение PCPT и предоставляемые вместе с книгой экзамены позволят сдавать пробные экзамены и ICND1, и CCNA.

При сдаче пробного экзамена ICND1 необходимо выбрать один или несколько экзаменов ICND1 в программном обеспечении PCPT. Если вы следовали плану изучения этой книги, то еще не видели ни одного из вопросов в этих базах данных экзаменов. После выбора одного из этих экзаменов достаточно выбрать параметр Practice Exam (Пробный экзамен) вверху справа и запустить экзамен.

Пробные экзамены можно использовать несколькими способами. Если выбран одиночный пробный экзамен, то можно провести четыре пробные сдачи, не встретив повторяющихся вопросов. В этом случае можно выделить все четыре экзамена, и программное обеспечение PCPT случайно выберет набор вопросов из всех четырех. При таком подходе можно сделать очень много попыток сдачи пробного экзамена, прежде чем вы начнете запоминать конкретные вопросы.

Табл. 30.3 содержит контрольный список для записи событий пробного экзамена. Обратите внимание, что примечания и дата в таблице кадров весьма полезны для некоторых других действий, поэтому заполняйте оба поля. Кроме того, в столбце примечаний, если задание выполнено раньше времени, отметьте количество оставшегося времени; если времени не хватило, отметьте количество вопросов, на которые не хватило времени.

Таблица 30.3. Контрольный список пробного экзамена ICND1

Экзамен	Дата	Результат	Примечания, время
ICND1			
ICND1			
ICND1			
ICND1			

Пробная сдача экзамена CCNA

Если решено сдать только экзамен CCNA, то и пробные экзамены нужно сдавать CCNA, а не ICND1 или ICND2. Комплект вопросов пробного экзамена CCNA используется пробными экзаменами ICND1 и ICND2, но лучше отвечать на те вопросы, которые будут на вашем экзамене.

И книга по ICND1, и книга по ICND2 предоставляет по четыре комплекта экзаменационных вопросов CCNA каждый. Если имеется только одна из этих двух книг, используйте четыре экзамена раздела "CCNA Full Exam" (Полный экзамен CCNA). Если есть обе книги, то имеется два комплекта по четыре экзамена CCNA, т.е. в общей сложности восемь индивидуальных экзаменов CCNA. На рис. 30.1 приведены названия и смысл экзаменов в программном обеспечении PCPT.

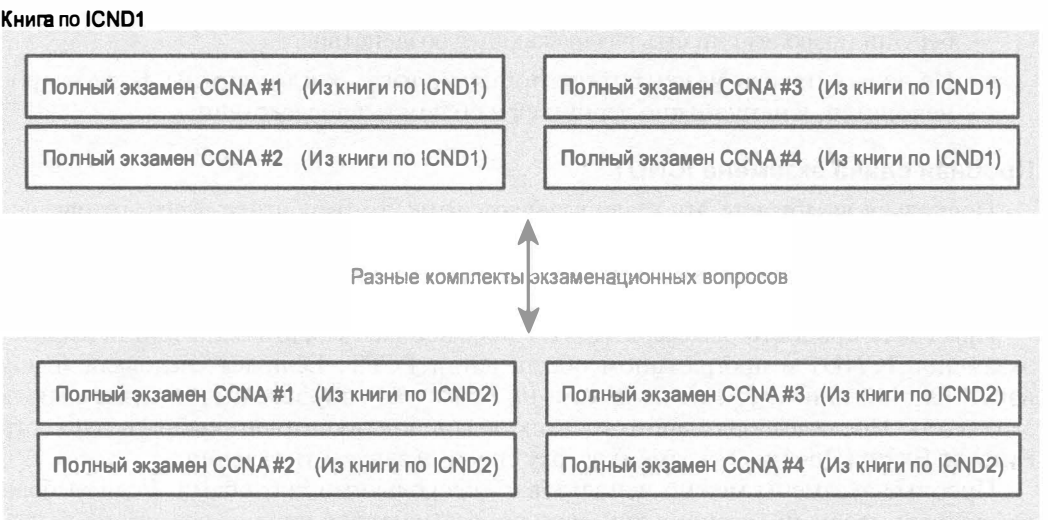


Рис. 30.1. Комплекты экзаменационных вопросов CCNA в книгах ICND1 и ICND2

Для сдачи экзамена CCNA выберите в окне PCPT одну из экзаменационных баз данных CCNA. Затем в разделе Practice Exam (Пробный экзамен) выберите параметр Mode (Режим) и запустите экзамен.

В табл. 30.4 приведен контрольный список для записи событий пробного экзамена. Обратите внимание, что примечания и дата в таблице кадров весьма полезны для некоторых других действий, поэтому заполняйте оба поля. Кроме того, в столбце примечаний, если задание выполнено раньше времени, отметьте количество ос-

тавшегося времени; если времени не хватило, отметьте количество вопросов, на которые не хватило времени.

Таблица 30.4. Контрольный список пробного экзамена CCNA

Название экзаменационной базы данных	Дата	Результат	Примечания, время
Экзамен CCNA 1 (из книги по ICND1)			
Экзамен CCNA 2 (из книги по ICND1)			
Экзамен CCNA 3 (из книги по ICND1)			
Экзамен CCNA 4 (из книги по ICND1)			
Экзамен CCNA 1 (из книги по ICND2)			
Экзамен CCNA 2 (из книги по ICND2)			
Экзамен CCNA 3 (из книги по ICND2)			
Экзамен CCNA 4 (из книги по ICND2)			

Советы о том, как отвечать на экзаменационные вопросы

Откройте веб-браузер. Да, отдохните и откройте веб-браузер на любом устройстве. Найдите интересную тему. Затем, прежде чем щелкнуть на ссылке, проследите, куда устремятся ваши глаза в течение первых 5–10 секунд после щелчка на ссылке. Потом щелкните на ссылке и посмотрите на страницу. Куда вы смотрите?

Интересно, что дизайн веб-браузеров и содержимого веб-страниц приучили всех к определенному стилю просмотра. Дизайнеры веб-страниц разрабатывают содержимое с учетом того, что люди просматривают страницы по разным шаблонам. Независимо от шаблона чтения веб-страниц, почти никто не читает их последовательно и полностью. Сначала люди просматривают интересующие их рисунки и заголовки, а затем то, что находится вокруг этих элементов.

Остальные средства электронной культуры также повлияли на способ восприятия информации средним человеком. Например, большинство людей, пользуясь социальными сетями и средствами текстовых сообщений, привыкли отсеивать интересное из сотен и даже тысяч сообщений, где каждое сообщение занимает целое предложение.

Эти каждодневные привычки повлияли на то, как все мы читаем и думаем перед экраном. К сожалению, те же привычки зачастую служат плохую службу при сдаче машинных экзаменов.

Если вы будете просматривать экзаменационные вопросы так же, как читаете веб-страницы, электронные письма и сообщения социальных сетей, то, вероятно, пропустите ключевой факт в вопросе, ответе или на рисунке. Учитесь внимательно читать все слова с начала и до конца, что для многих людей стало на удивление противоестественным.

ВНИМАНИЕ!

Автор говорил со многими университетскими профессорами по разным дисциплинам, а также с преподавателями академии Cisco Networking Academy, и все они в один голос заявляли, что главной проблемой на экзаменах является то, что люди не читают вопросы достаточно внимательно, чтобы понять подробности.

Позвольте автору дать две рекомендации о сдаче пробных экзаменов и ответе на отдельные вопросы. Во-первых, перед пробным экзаменом обдумайте собственную стратегию чтения вопросов. Выработайте собственный подход к вопросам, в особенности к вопросам с многовариантным выбором нескольких ответов. Во-вторых, если хотите несколько рекомендаций о чтении экзаменационного вопроса, используйте следующую стратегию.

Этап 1 Прочитайте вопрос полностью, от начала до конца

Этап 2 Просмотрите все дополнения (обычно вывод команды) и рисунки

Этап 3 Просмотрите ответы, чтобы понять тип информации. (Числа? Термины? Отдельные слова? Фразы?)

Этап 4 Еще раз перечитайте вопрос полностью, от начала до конца, чтобы удостовериться в его понимании

Этап 5 Прочитайте каждый ответ полностью, обращая внимание на рисунки, если они есть. После чтения каждого ответа, прежде чем приступить к чтению следующего ответа:
A. Если точно правильный, выберите его.
B. Если ответ наверняка неправилен, мысленно исключите его.
C. Если не уверены, мысленно отметьте ответ как возможно правильный

ВНИМАНИЕ!

Количество правильных ответов на экзамене Cisco указано. Экзаменационное программное обеспечение также поможет закончить вопрос с правильным количеством ответов. Оно не позволит выбрать слишком много ответов. Кроме того, если вы попытаетесь перейти к следующему вопросу, выделив слишком мало ответов, экзаменационное программное обеспечение переспросит, действительно ли вы хотите это сделать.

Используйте пробные экзамены, чтобы опробовать свой подход к чтению. Переходя к каждому следующему вопросу, попытайтесь читать его согласно выбранному подходу. Если чувствуете нехватку времени, то нужно продолжить практиковать свой подход, сокращая количество вопросов, пропущенных только из-за просмотра, а не полного чтения.

Обзор вопросов поможет найти пробелы в знаниях

Вы только что прошли несколько пробных экзаменов, вероятно, многому научились, извлекли некоторую пользу для себя, приобрели навыки и улучшили знание сети. Но если вернуться и просмотреть все вопросы без правильных ответов, то можно обнаружить несколько небольших пробелов в знании.

Одной из труднейших задач при окончательной подготовке к экзамену является обнаружение пробелов в знаниях и навыках. Другими словами, необходимо узнать, какими темами и навыками вы владеете слабо? Или какие темы вы полагаете известными, но неправильно понимаете некоторые важные факты? Поиск пробелов в своих знаниях на данном последнем рабочем этапе требует больше, чем просто знания своих сильных и слабых сторон.

Программное обеспечение PCPT поможет найти эти пробелы; оно отслеживает каждый сданный пробный экзамен, запоминая ответ на каждый вопрос и правильность ответа. Просматривая результаты, можно перемещаться между вопросами и просматривать страницы с результатами. Для поиска пробелов в своих знаниях следуйте таким этапам.

Этап 1 Выберите и просмотрите один из пробных экзаменов

Этап 2 Просматривайте все вопросы с неправильными ответами, пока не будете уверены в правильном понимании вопросов

Этап 3 Завершив обзор вопроса, отметьте его

Этап 4 Продолжайте обзор всех вопросов с неправильными ответами, пока все они не будут отмечены.

Этап 5 Переходите к следующему пробному экзамену

На рис. 30.2 приведена типичная страница из обзора вопросов, которые имеют неправильные ответы. Результаты перечислены в столбце Correct (Правильно), отсутствие флажка означает неправильный ответ.

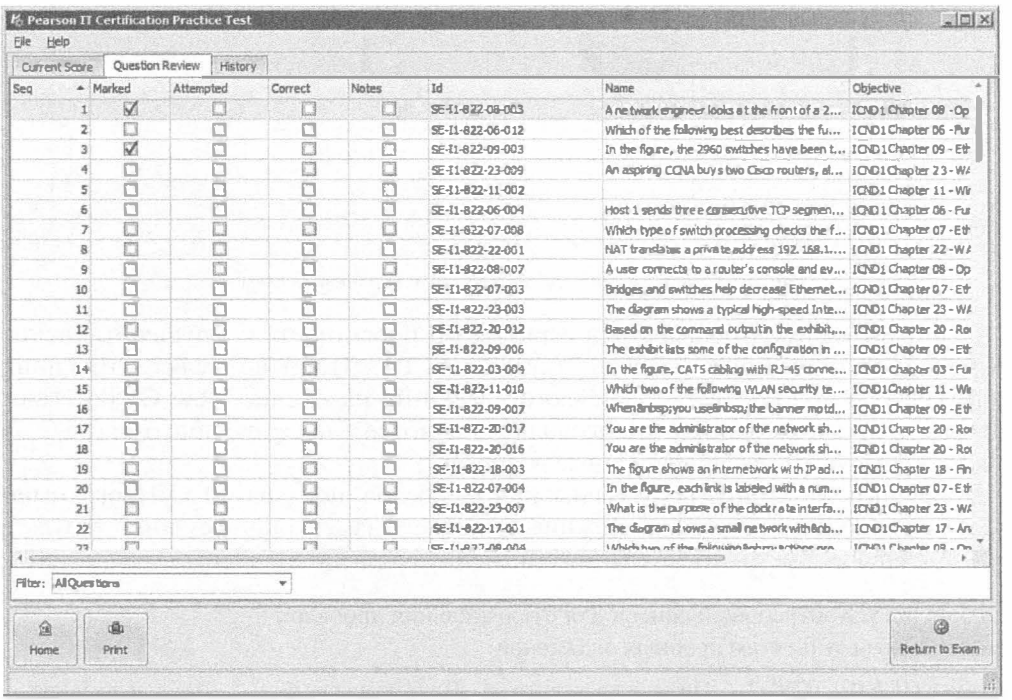


Рис. 30.2. Аттестационная страница PCPT с результатами

В процессе обзора вопросов и отметки их выполнения можно перемещаться между страницей обзора всех вопросов и индивидуальными вопросами. Достаточно дважды щелкнуть на вопросе, чтобы отобразить его. В окне вопроса можно щелкнуть на кнопке Grade Exam (Оценка экзамена), чтобы перейти к результатам аттестации на странице обзора вопросов, представленной на рис. 30.2. В окне вопроса есть также возможность отметить вопрос (в верхнем левом углу), как показано на рис. 30.3.

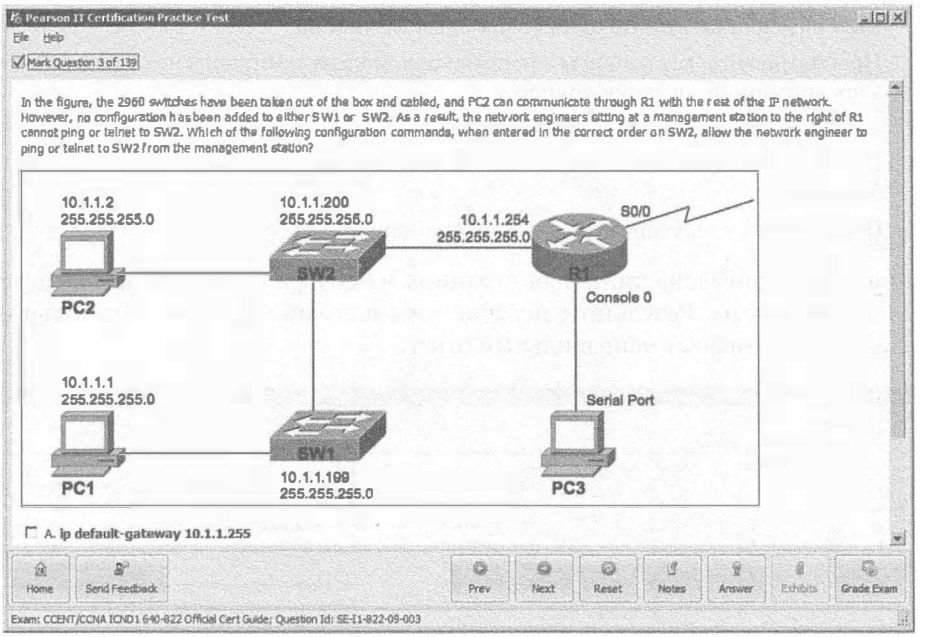


Рис. 30.3. Обзор вопроса с отметкой в верхнем левом углу

Если впоследствии понадобится вернуться и просмотреть пропущенные вопросы, это можно сделать на начальном экране PCPT. Чтобы не щелкать на кнопке Start (Пуск) и не открывать новое окно, щелкните на кнопке View Grade History (Просмотр истории оценок). Это позволит просмотреть прежние попытки сдачи экзамена и обработать все пропущенные вопросы.

Проследите свой прогресс выявления пробелов по табл. 30.5. Программное обеспечение PCPT отображает прежние пробные экзамены по времени и результату, — так проще заметить необходимые значения согласно таблице.

Таблица 30.5. Контрольный список для отслеживания пробелов в знаниях по результатам пробных экзаменов

Экзамен (ICND1, ICND2 или CCNA)	Первоначальная дата пробного экзамена	Первоначальный результат экзамена	Дата ликвидации пробела

Практические навыки CLI

Для успешного ответа на вопросы с симлетами необходимы навыки работы с командами маршрутизаторов и коммутаторов Cisco, а также использования CLI Cisco. Как было сказано во введении к этой книге, вопросы симлетов требуют при-

нятия решения, какие команды конфигурации следует ввести, чтобы решить проблему или закончить рабочую конфигурацию. Вопросы симлетов требуют ответить на вопросы с множественным выбором, используя предварительно CLI для ввода команды `show`, позволяющие просмотреть состояние маршрутизаторов и коммутаторов в небольшой сети.

При подготовке к экзамену необходимо знать следующие виды информации.

- **Навигация CLI**. Базовый механизм CLI перемещения между режимами пользователя, привилегированным режимом и режимом конфигурации.
- **Индивидуальная конфигурация**. Смысл параметров каждой команды конфигурации.
- **Конфигурация средства**. Набор команд конфигурации, обязательных и необязательных, для каждого средства.
- **Проверка конфигурации**. Команды `show`, непосредственно идентифицирующие параметры конфигурации.
- **Проверка состояния**. Команды `show`, выводящие текущее состояние и позволяющие выявить ошибки конфигурации или другие причины проблем по отличным от оптимальных значениям состояния.

Чтобы лучше усвоить эти знания и навыки, можно выполнить задания, перечисленные в нескольких следующих разделах.

Диаграммы связей из обзоров частей

В упражнениях обзоров частей вы создавали разные диаграммы связей и по конфигурации, и по командам проверки. Чтобы запомнить конкретные диаграммы связей, вернитесь к разделам обзоров каждой части.

Выполнение лабораторных работ

Вне зависимости от выбранного метода приобретения практических навыков работы с CLI, уделите время обзору и выполнению лабораторных работ по командам. На настоящий момент вы имеете немного практических навыков ввода команд конфигурации, полученных при тренировке на эмуляторе, реальном механизме или на бумаге. Хотя повторение всех лабораторных работ могло бы быть и непрактично, тренировка с любыми командами и средствами, в которых вы чувствуете себя немного неуверенным, а также по темам из обзора диаграмм связей имела бы смысл. Контрольный список лабораторных работ приведен в табл. 30.6.

Таблица 30.6. Контрольный список лабораторных работ

Тема	Глава	Дата выполнения лабораторной работы
Основы CLI: пароли, имена хостов, баннеры и т.д.	7	
Коммутатор Ipv4	8	
Защита порта коммутатора	8	
VLAN	9	
Магистральное соединение VLAN	9	

Окончание табл. 30.6

Тема	Глава	Дата выполнения лабораторной работы
IPv4-адреса маршрутизатора и статические маршруты	16	
OSPFv2	17	
Стандартные списки доступа	22	
Расширенные и именованные списки ACL	23	
NAT	24	
Адресация IPv6 на маршрутизаторах	27	
Статические маршруты IPv6	29	
OSPFv3 (для IPv6)	29	

Наилучший способ приобретения практических навыков заключается в использовании эмулятора Pearson Network Simulator (или Sim), см. `pearsonitcertification.com/networksimulator`.

В качестве бесплатной альтернативы можно выполнить на бумаге несколько коротких, на 5–10 минут, лабораторных работ по конфигурации, приведенных на блогах автора. Найдите их в разделе Config Museum блогов (один блог по ICND1, а второй по ICND2) и выберите те лабораторные работы, которые хотите использовать. Можете попробовать выполнить их на бумаге или в собственном средстве выполнения лабораторной работы. Начать поиск блогов можно по адресу `www.certskills.com/blogs`.

Другие учебные задачи

Если, дойдя до этого места, вы все еще чувствуете потребность в некой подготовке, то этот последний раздел предоставляет еще две рекомендации.

Во-первых, некоторые полезные задачи предоставляют разделы обзоров глав и частей. Во-вторых, примите участие в обсуждениях учебной сети Cisco Learning Network. Попробуйте отвечать на вопросы, задаваемые другими участниками; процесс ответа заставляет обдумать тему намного глубже. Когда некто публикует ответ, с которым вы не согласны, обдумайте причину и выскажите свое мнение. Это отличный способ узнать больше и ощутить атмосферу доверия.

Заключительные соображения

Вы много учились, упорно трудились и потратили время и деньги на подготовку к экзамену. Надеюсь, экзамен пройдет успешно, поскольку вы действительно знаете материал и преуспеете в карьере сетевого специалиста.

Празднуйте успех и не расстраивайтесь в противном случае. Учебная сеть Cisco Learning Network является прекрасным местом, чтобы публиковать сообщения и просить советы на следующий раз. Автор лично хотел бы услышать о вашем успехе через Twitter (`@wendellodom`) или на его странице в Facebook (`facebook.com/wendellodom`). Желаю вам успеха и поздравляю с завершением работы над книгой!

Часть IX. Приложения

Приложение А. “Справочные числовые таблицы”

Приложение Б. “Обновление экзамена ICND1”

Список терминов

Справочные числовые таблицы

Это приложение содержит несколько полезных справочных таблиц, в которых приведены числа, используемые всюду в этой книге. Например, табл. А.1 полезна при преобразовании десятичных чисел в двоичные, и наоборот.

Таблица А.1. Десятичные и двоичные числа в диапазоне от 0 до 255

Десятичное число	Двоичное число	Десятичное число	Двоичное число	Десятичное число	Двоичное число	Десятичное число	Двоичное число
0	00000000	32	00100000	64	01000000	96	01100000
1	00000001	33	00100001	65	01000001	97	01100001
2	00000010	34	00100010	66	01000010	98	01100010
3	00000011	35	00100011	67	01000011	99	01100011
4	00000100	36	00100100	68	01000100	100	01100100
5	00000101	37	00100101	69	01000101	101	01100101
6	00000110	38	00100110	70	01000110	102	01100110
7	00000111	39	00100111	71	01000111	103	01100111
8	00001000	40	00101000	72	01001000	104	01101000
9	00001001	41	00101001	73	01001001	105	01101001
10	00001010	42	00101010	74	01001010	106	01101010
11	00001011	43	00101011	75	01001011	107	01101011
12	00001100	44	00101100	76	01001100	108	01101100
13	00001101	45	00101101	77	01001101	109	01101101
14	00001110	46	00101110	78	01001110	110	01101110
15	00001111	47	00101111	79	01001111	111	01101111
16	00010000	48	00110000	80	01010000	112	01110000
17	00010001	49	00110001	81	01010001	113	01110001
18	00010010	50	00110010	82	01010010	114	01110010
19	00010011	51	00110011	83	01010011	115	01110011
20	00010100	52	00110100	84	01010100	116	01110100
21	00010101	53	00110101	85	01010101	117	01110101
22	00010110	54	00110110	86	01010110	118	01110110
23	00010111	55	00110111	87	01010111	119	01110111
24	00011000	56	00111000	88	01011000	120	01111000
25	00011001	57	00111001	89	01011001	121	01111001

Окончание табл. А.1

Десятичное число	Двоичное число	Десятичное число	Двоичное число	Десятичное число	Двоичное число	Десятичное число	Двоичное число
26	00011010	58	00111010	90	01011010	122	01111010
27	00011011	59	00111011	91	01011011	123	01111011
28	00011100	60	00111100	92	01011100	124	01111100
29	00011101	61	00111101	93	01011101	125	01111101
30	00011110	62	00111110	94	01011110	126	01111110
31	00011111	63	00111111	95	01011111	127	01111111
128	10000000	160	10100000	192	11000000	224	11100000
129	10000001	161	10100001	193	11000001	225	11100001
130	10000010	162	10100010	194	11000010	226	11100010
131	10000011	163	10100011	195	11000011	227	11100011
132	10000100	164	10100100	196	11000100	228	11100100
133	10000101	165	10100101	197	11000101	229	11100101
134	10000110	166	10100110	198	11000110	230	11100110
135	10000111	167	10100111	199	11000111	231	11100111
136	10001000	168	10101000	200	11001000	232	11101000
137	10001001	169	10101001	201	11001001	233	11101001
138	10001010	170	10101010	202	11001010	234	11101010
139	10001011	171	10101011	203	11001011	235	11101011
140	10001100	172	10101100	204	11001100	236	11101100
141	10001101	173	10101101	205	11001101	237	11101101
142	10001110	174	10101110	206	11001110	238	11101110
143	10001111	175	10101111	207	11001111	239	11101111
144	10010000	176	10110000	208	11010000	240	11110000
145	10010001	177	10110001	209	11010001	241	11110001
146	10010010	178	10110010	210	11010010	242	11110010
147	10010011	179	10110011	211	11010011	243	11110011
148	10010100	180	10110100	212	11010100	244	11110100
149	10010101	181	10110101	213	11010101	245	11110101
150	10010110	182	10110110	214	11010110	246	11110110
151	10010111	183	10110111	215	11010111	247	11110111
152	10011000	184	10111000	216	11011000	248	11111000
153	10011001	185	10111001	217	11011001	249	11111001
154	10011010	186	10111010	218	11011010	250	11111010
155	10011011	187	10111011	219	11011011	251	11111011
156	10011100	188	10111100	220	11011100	252	11111100
157	10011101	189	10111101	221	11011101	253	11111101
158	10011110	190	10111110	222	11011110	254	11111110
159	10011111	191	10111111	223	11011111	255	11111111

В табл. А.2 приведены шестнадцатеричные и двоичные числа. Она полезна при преобразовании шестнадцатеричных чисел в двоичные, и наоборот.

Таблица А.2. Шестнадцатеричные и двоичные числа

Шестнадцатеричное число	Четырехзначное двоичное число
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Табл. А.3 содержит степени числа 2, от 2^1 до 2^{32}.

Таблица А.3. Степени числа 2

X	2^X	X	2^X
1	2	17	131 072
2	4	18	262 144
3	8	19	524 288
4	16	20	1 048 576
5	32	21	2 097 152
6	64	22	4 194 304
7	128	23	8 388 608
8	256	24	16 777 216
9	512	25	33 554 432
10	1024	26	67 108 864
11	2048	27	134 217 728
12	4096	28	268 435 456
13	8192	29	536 870 912
14	16 384	30	1 073 741 824
15	32 768	31	2 147 483 648
16	65 536	32	4 294 967 296

В табл. А.4 приведены все 33 возможные маски подсетей, во всех трех форматах.

Таблица А.4. Все маски подсетей

Десятичная	Префикс	Двоичная
0.0.0.0	/0	00000000 00000000 00000000 00000000
128.0.0.0	/1	10000000 00000000 00000000 00000000
192.0.0.0	/2	11000000 00000000 00000000 00000000
224.0.0.0	/3	11100000 00000000 00000000 00000000
240.0.0.0	/4	11110000 00000000 00000000 00000000
248.0.0.0	/5	11111000 00000000 00000000 00000000
252.0.0.0	/6	11111100 00000000 00000000 00000000
254.0.0.0	/7	11111110 00000000 00000000 00000000
255.0.0.0	/8	11111111 00000000 00000000 00000000
255.128.0.0	/9	11111111 10000000 00000000 00000000
255.192.0.0	/10	11111111 11000000 00000000 00000000
255.224.0.0	/11	11111111 11100000 00000000 00000000
255.240.0.0	/12	11111111 11110000 00000000 00000000
255.248.0.0	/13	11111111 11111000 00000000 00000000
255.252.0.0	/14	11111111 11111100 00000000 00000000
255.254.0.0	/15	11111111 11111110 00000000 00000000
255.255.0.0	/16	11111111 11111111 00000000 00000000
255.255.128.0	/17	11111111 11111111 10000000 00000000
255.255.192.0	/18	11111111 11111111 11000000 00000000
255.255.224.0	/19	11111111 11111111 11100000 00000000
255.255.240.0	/20	11111111 11111111 11110000 00000000
255.255.248.0	/21	11111111 11111111 11111000 00000000
255.255.252.0	/22	11111111 11111111 11111100 00000000
255.255.254.0	/23	11111111 11111111 11111110 00000000
255.255.255.0	/24	11111111 11111111 11111111 00000000
255.255.255.128	/25	11111111 11111111 11111111 10000000
255.255.255.192	/26	11111111 11111111 11111111 11000000
255.255.255.224	/27	11111111 11111111 11111111 11100000
255.255.255.240	/28	11111111 11111111 11111111 11110000
255.255.255.248	/29	11111111 11111111 11111111 11111000
255.255.255.252	/30	11111111 11111111 11111111 11111100
255.255.255.254	/31	11111111 11111111 11111111 11111110
255.255.255.255	/32	11111111 11111111 11111111 11111111

Обновление экзамена ICND1

Отзывы читателей помогают издательству Cisco Press определить, какие именно темы вызывают наибольшие сложности на сертификационном экзамене. Более того, компания Cisco может постепенно вносить небольшие изменения в темы экзаменов и по-другому расставлять акценты и приоритеты в технологиях передачи данных. Чтобы помочь читателю в работе над изменившимися темами, автор книги публикует дополнительные материалы, в которых объяснены трудные моменты каких-либо технологий и новых тем экзамена.

Данное приложение имеет версию 1.0 и не содержит никаких обновлений. Проверить модифицированную версию можно по адресу `www.ciscopress.com/title/9781587144851`.

Список терминов

1000BASE-T. Спецификация широкополосной технологии Gigabit Ethernet со скоростью передачи 1000 Мбит/с, в которой используются четыре пары кабеля UTP категории 5; максимальная длина сегмента составляет 100 м (328 футов).

100BASE-TX. Спецификация узкополосной технологии Fast Ethernet со скоростью передачи данных 100 Мбит/с, в которой используются две пары кабелей UTP или STP. Первая пара используется для приема данных, вторая — для передачи. Для нормальной синхронизации сигнала в 100BASE-TX соединение не должно превышать 100 м (328 футов). Описывается стандартом IEEE 802.3.

10BASE-T. Спецификация узкополосной технологии Ethernet со скоростью передачи данных 10 Мбит/с, в которой используются две пары кабеля типа "витая пара" (категории 3, 4, 5): одна пара — для передачи данных, другая — для приема. Она является частью стандарта IEEE 802.3; максимальная длина сегмента равна 100 м (328 футов).

802.11a. Стандарт IEEE для беспроводных сетей с использованием спектра U-NII, с модуляцией OFDM и скоростью до 54 Мбит/с.

802.11b. Стандарт IEEE для беспроводных сетей с использованием спектра ISM, с модуляцией DSSS и скоростью до 11 Мбит/с.

802.11g. Стандарт IEEE для беспроводных сетей с использованием спектра ISM, с модуляцией DSSS и скоростью до 54 Мбит/с.

802.11i. Стандарт IEEE безопасности беспроводных сетей, включающий в себя аутентификацию и шифрование.

802.11n. Стандарт IEEE для беспроводных локальных сетей с использованием спектра ISM, модуляцией OFDM и нескольких антенн для одиночного потока со скоростью до 150 Мбит/с.

802.1Q. Стандарт IEEE для магистральных каналов сетей VLAN.

Anti-X. Термин, используемый компанией Cisco для описания многих средств сетевой безопасности, предотвращающих различные атаки, включая антивирус, антифишинг и антиспам.

ARPANET. Первая сеть с коммутацией пакетов, впервые созданная в 1970-х годах. Предшественница Интернета.

E1. Цифровой канал передачи данных с полосой 2,048 Мбит/с, состоящий из 32 подканалов по 64 Кбит/с, из которых один зарезервирован для фреймирования и служебных данных. Используется в Европе и является аналогом американского стандарта T1.

EtherType. Жаргонное сокращение термина "Ethernet Type", обозначающего поле Type в заголовке Ethernet. Поле Type идентифицирует тип пакета, инкапсулируемого во фрейме Ethernet.

Fast Ethernet. Общее название для всех стандартов IEEE передачи данных на скорости 100 Мегабит в секунду.

Gigabit Ethernet. Общее название для всех стандартов IEEE передачи данных на скорости 1 Гигабит в секунду.

IP-адрес (версия 4) (IP address (IP version 4)). В версии 4 (IPv4) это 32-битовый адрес, назначаемый хосту при использовании протокола TCP/IP. Каждый адрес состоит из адреса сети, необязательного адреса подсети и хоста. Адреса сети и подсети совместно используются для маршрутизации, а адрес хоста необходим для доставки информации определенному сетевому хосту в сети или подсети. Маска подсети используется для извлечения из IP-адреса информации о сети и подсети.

IP-адрес (версия 6) (IP address (IP version 6)). В версии 6 (IPv6) 128-битовый адрес, назначаемый хосту при использовании протокола TCP/IP. Адреса используют различные форматы, обычно используется префикс маршрутизации, идентификатор подсети и интерфейса, соответствующие частям сети, подсети и хоста IPv4-адреса.

L4PDU. Блок данных протокола уровня 4., содержащий заголовок четвертого уровня и инкапсулированные данные верхних уровней, но не информацию нижних.

MAC-адрес (MAC address). Стандартизованный адрес канального уровня, необходимый каждому устройству, подключенному к локальной сети. Все устройства используют MAC-адреса, чтобы найти определенные устройства в сети, а также для создания и обновления таблиц коммутации и структур данных. Длина MAC-адресов составляет 6 байтов, контролируются они Институтом инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers — IEEE). Этот тип адреса также называют *аппаратным адресом* (hardware address), *адресом уровня MAC* (MAC-layer address) и *физическим адресом* (physical address).

RJ-45. Разъем с восемью контактами, обычно заканчивающий кабель типа "витая пара" в сети Ethernet. Похож на телефонный разъем RJ-11, весьма распространенный в США и Европе.

T1. Цифровой канал передачи данных по распределенной сети. В линии T1 данные передаются в формате DS-1 со скоростью 1,544 Мбит/с в виде 24 отдельных подканалов по 64 Кбит/с (DS0) и управляющего канала для служебных данных со скоростью 8 Кбит/с.

Telco. Общепринятое англоязычное сокращение от слов "телефонная компания".

Traceroute (сокращенный вариант — trace). Программа, которая прослеживает путь пакета до пункта назначения. Используется главным образом для отладки процесса маршрутизации между хостами. Существует также протокол отслеживания, определенный в документе RFC 1393. Эта программа имеется во многих операционных системах.

Абонентский канал (local loop). Телефонная линия или телекоммуникационный канал от оборудования абонента до АТС оператора местной телефонной связи.

Автоматическая настройка адреса (Stateless Address Autoconfiguration — SLAAC) (). Средство протокола IPv6, позволяющее хосту или маршрутизатору присвоить одноадресатный IPv6-адрес без потребности в фиксации состояния на сервере DHCP.

Автономная система (autonomous system). Отдельная сеть или набор сетей, находящихся под единым административным контролем какой-либо компании, государственной организации. В автономной системе обычно используется *протокол маршрутизации внутреннего шлюза* (Interior Gateway Protocol — IGP).

Автопереговоры (autonegotiation). Механизм стандарта IEEE (802.3u), позволяющий двум узлам обмениваться сообщениями о выборе одинакового стандарта Ethernet на обоих концах канала связи. Гарантирует корректное функционирование канала связи.

Авторизация (authorization). В технологиях безопасности проверка прав определенного пользователя или устройства. См. *AAA*.

Административный режим магистрали (trunking administrative mode). Настраиваемый параметр магистрального соединения на интерфейсе коммутатора Cisco, задаваемый командой **`switchport mode`**.

Адрес Ethernet (Ethernet address). 48-битовое (6-байтовое) двоичное число, обычно записываемое как шестнадцатеричное число с 12 цифрами и используемое для идентификации узлов в сети Ethernet. В заголовке фрейма Ethernet содержатся поля адресов получателя и отправителя, используемые устройствами Ethernet для доставки фреймов правильному получателю.

Адрес подсети (subnet number или subnet address). В протоколе IPv4 — это 4-байтовое число, записываемое в десятичной форме с точками между байтами, которое описывает все адреса в подсети. В числовом выражении — это наименьший адрес в подсети, который является зарезервированным и не может быть назначен хосту.

Адрес хоста (host address). IP-адрес, присвоенный сетевой карте компьютера.

Адрес, локальный в пределах канала связи (link-local address). Тип одноадресатного IPv6-адреса, представляющего интерфейс на одном канале связи. Посланные на этот адрес пакеты передаются только в пределах конкретного канала связи и никогда не передаются маршрутизатором в другие подсети. Используется для сообщений, которые не должны покидать локальный канал связи.

Алгоритм выбора первого кратчайшего маршрута (Shortest Path First Algorithm — SPF). Алгоритм маршрутизации, используемый в протоколах маршрутизации по состоянию канала для анализа базы LSDB и поиска наименее дорогого маршрута от данного маршрутизатора до каждой подсети.

Альянс Wi-Fi (Wi-Fi Alliance). Организация, сформированная множеством компаний, производящих оборудование для беспроводных сетей (т.е. промышленная ассоциация), основная задача которой — сформировать рынок совместимых между собой устройств от разных производителей. Эта организация занимается выпуском стандартов и ускоряет процесс стандартизации в международных организациях.

Анонс маршрутизации (routing update). Сообщения, рассылаемые между маршрутизаторами объединенной сети, в которых содержится информация о достижимости сети и соответствующая оценка маршрута. Обновления маршрутизации обычно рассылаются с постоянными интервалами, а также в случае изменений в сетевой топологии.

Анонс соседа (Neighbor Advertisement — NA). Сообщение, определенное протоколом обнаружения соседних устройств IPv6 (NDP), используемое для объявления другим соседям о MAC-адресе хоста. Иногда посылается в ответ на ранее полученный запрос соседа (NS).

Анонс состояния канала (Link-State Advertisement — LSA). Структура данных протокола OSPF в базе LSDB, где описаны детали разных компонентов сети, в том числе маршрутизаторов и каналов (подсетей).

Анонсирование маршрутизатора (Router Advertisement — RA). Сообщение, определенное протоколом обнаружения соседних устройств IPv6 (NDP), используемое маршрутизаторами для объявления о готовности действовать как маршрутизатор IPv6 на канале связи. Может быть послано в ответ на ранее полученный запрос.

Асимметричный цифровой абонентский канал (Asymmetric Digital Subscriber Line — ADSL). Одна из четырех технологий DSL, предназначенная для высокоскоростной передачи данных в направлении основного трафика (от центрального офиса к пользователю), а не в обратном направлении. Канал ADSL функционирует на расстоянии до 18 000 футов (5488 метров) по одиночной электрической витой паре.

Асимметрия (asymmetric). Свойство многих технологий доступа к Интернету, в том числе и DSL, предполагающее, что скорость входящего потока данных намного больше, чем исходящего.

Асинхронность (asynchronous). Отсутствие временных меток в потоке битов. На практике оба конца канала согласовывают одинаковую скорость передачи данных, но в дальнейшем подстройка или проверка того факта, что скорости передачи немного отличаются, отсутствует. Тем не менее, поскольку данные пересылаются побитово, небольшая рассинхронизация не является проблемой.

Асинхронный режим передачи (Asynchronous Transfer Mode — ATM). Международный стандарт поэлементной передачи, при которой несколько типов данных (например, голосовые, цифровые и видеоданные) передаются в виде ячеек фиксированной длины (53 байта). Ячейки фиксированной длины обрабатываются на аппаратном уровне, что, в свою очередь, позволяет сократить задержки при передаче. Режим ATM предназначен для таких высокоскоростных сетей, как E3, SONET и T3.

Аутентификация (authentication). В технологиях безопасности — проверка идентификатора пользователя или процесса.

Аутентификация, авторизация и учет (Authentication, Authorization, And Accounting — AAA). Произносится как "triple a". С помощью аутентификации идентифицируют пользователя или устройство. Авторизация определяет права на выполнение чего-либо пользователем, а учет записывает информацию о попытках доступа.

База данных состояний каналов (Link-State Database — LSDB). Структура данных в оперативной памяти маршрутизатора OSPF, в которой хранятся анонсы LSA, используемые для построения полной топологии сети.

Базовый набор служб (Basic Service Set — BSS). Беспроводная сеть с единственной точкой доступа к сети.

Без контроля ошибок (error disabled). Специальное состояние интерфейса в коммутаторе локальной сети, в которое он переходит из-за различных нарушений режима безопасности.

Бесклассовая междоменная маршрутизация (Classless InterDomain Routing — CIDR). Определенный документом RFC стандартный инструмент присвоения диапазонов глобальных IP-

адресов. Бесклассовая междоменная маршрутизация сокращает размер таблиц маршрутизации IP маршрутизаторов Интернета, позволяя справиться с быстрым ростом Интернета. Термин *бесклассовый* указывает на тот факт, что полученные в итоге группы сетей представляют группы адресов, никак не соответствующие правилам группировки классовых сетей IPv4 (классам A, B и C).

Бесклассовый протокол маршрутизации (classless routing protocol). Более новый протокол маршрутизации, пересылающий в своих анонсах таблиц маршрутизации адрес подсети и маску. Такой протокол маршрутизации не ориентирован на класс сети и поддерживает маски VLSM и суммирование маршрутов вручную.

Беспроводная локальная сеть (wireless LAN). Локальная сеть (LAN), физически передающая биты по радиоканалу. Название "беспроводная" противопоставляет эти локальные сети традиционным "проводным", использующим кабели (как правило, с медными проводами внутри).

Блок адресов (address block). Набор последовательных адресов протоколов IPv4 и IPv6. Как правило, термин используется для открытых адресов, присваиваемых некой уполномоченной организацией (IANA/ICANN, RIR или ISP).

Блок данных протокола (Protocol Data Unit — PDU). Термин в модели OSI, описывающий сгруппированную определенным образом информацию какого-либо конкретного уровня модели. Обычно аббревиатурой L*x*PDU обозначают блок уровня *x* (Layer *x*).

Брандмауэр (firewall). Одно или несколько сетевых устройств, таких как маршрутизаторы или серверы доступа, предназначенных для создания буферной зоны между открытыми и частными сетями. Для обеспечения безопасности частных сетей в брандмауэре используются списки управления доступом и другие методы.

Булево "И" (Boolean AND). Математическая операция для однобитового двоичного числа. В результате получается однобитовое число. 1 "И" 1 = 1, все остальные комбинации битов дают в результате 0.

Веб-сервер (web server). Программное обеспечение, запущенное на некотором компьютере, позволяющее хранить веб-страницы и передавать их клиентскому программному обеспечению — браузеру.

Взаимодействие на равноправном уровне (same-layer interaction). Коммуникация между двумя устройствами с использованием функций определенного уровня сетевой модели. Коммуникация осуществляется с использованием заголовков конкретного уровня. Например, устройство заполняет определенные поля в заголовках, пересылает заголовки и инкапсулированные данные, а принимающая сторона интерпретирует информацию в заголовках и выполняет соответствующие действия.

Взаимодействие на смежных уровнях (adjacent-layer interaction). Общий термин, описывающий, как два смежных уровня одного компьютера взаимодействуют в рамках сетевой модели. Нижний уровень предоставляет некоторую службу верхнему уровню.

Виртуальная локальная сеть (Virtual LAN — VLAN). Группа устройств, принадлежащих одной или нескольким локальным сетям и настроенных таким образом (с помощью управляющего программного обеспечения), что обмен данными между ними происходит так, как будто они подключены к одному кабелю, хотя на самом деле находятся в разных сегментах сети LAN. Поскольку сети VLAN основаны на логическом, а не на физическом соединении, они необычайно гибки.

Виртуальная частная сеть (Virtual Private Network — VPN). Частная сеть, создаваемая в открытой сетевой инфраструктуре, такой, например, как глобальный Интернет. В сетях VPN пакеты шифруются, поэтому обеспечивается конфиденциальность передаваемых данных, а оконечные точки сети аутентифицируются, что обеспечивает их идентичность.

Виртуальный канал (Virtual Circuit — VC). Логический канал, обеспечивающий надежное соединение между двумя сетевыми устройствами. Виртуальные каналы применяются в сетях Frame Relay, X.25 и ATM, обеспечивая те же функции, что и выделенная линия, но без выделенного физического соединения.

Витая пара (twisted pair). Среда передачи данных, представляющая собой два свитых медных провода в пластиковой оболочке без металлизированного экрана. За счет того, что проводники перекручены, интерференция и перекрестные помехи существенно уменьшаются. Широко используется в различных сетях.

Внутренний глобальный адрес (inside global). Заменяет существующий адрес в заголовках пакетов, передающихся из локальной сети (т.е. из-за устройства NAT) и использующихся для маршрутизации таких пакетов в глобальном (открытом) Интернете.

Внутренний локальный адрес (inside local). Адрес в заголовках пакетов, находящихся за устройством NAT в локальной корпоративной (т.е. частной) сети, подлежащий замене при передаче в открытую сеть.

Внутренний протокол маршрутизации (interior routing protocol). Протокол маршрутизации, предназначенный для использования в одной организации.

Восстановление после ошибок (error recovery). Процесс, позволяющий обнаружить, что данные были переданы с ошибками, и отвечающий за повторную пересылку недостающих или ошибочных блоков.

Вспомогательный порт (auxiliary port). Физический разъем маршрутизатора, предназначенный для подключения дистанционного терминала, например, компьютера с запущенной программой эмуляции терминала, соединяющегося с маршрутизатором через модем.

Выделенная линия (leased line). Разновидность последовательного канала связи между двумя точками без коммутации, зарезервированный поставщиком услуги, обычно — местной телефонной компанией, исключительно для использования заказчиком. Поскольку оператор связи или телефонная компания обычно не выделяет физический кабель между двумя площадками, а предоставляет виртуальный канал, стоимость такой линии может быть ниже.

Высокоуровневый протокол управления каналом (High-Level Data Link Control — HDLC). Бит-ориентированный синхронный протокол канального уровня, разработанный ISO. Основан на протоколе SDLC и определяет метод инкапсуляции данных в синхронных последовательных каналах с помощью символов кадрирования и контрольных сумм.

Глобальный одноадресатный адрес (global unicast address). Разновидность одноадресатного IPv6-адреса, относящегося к диапазону открытых глобально уникальных IP-адресов, зарегистрированного в Ассоциации IANA/ICANN, ее региональных представительствах или у крупных провайдеров услуг Интернета.

Глобальный префикс маршрутизации (global routing prefix). Префикс IPv6, определяющий блок IPv6-адреса, состоящий из присвоенных одной организации глобальных одноадресатных адресов с целью создания блока глобально уникальных IPv6-адресов для использования в ее сети.

Головной узел (head end). Вышестоящий узел абонентского канала кабельного телевидения для доступа к Интернету.

Граничный маршрутизатор зоны (Area Border Router — ABR). Маршрутизатор, интерфейсы которого используют протокол OSPF на нескольких площадках.

Двойной стек (dual stack). Метод работы устройств по протоколу IPv6, когда на маршрутизаторе одновременно запущены протоколы IPv6 и IPv4.

Декапсуляция (decapsulation), *или деинкапсуляция* (de-encapsulation). Процесс интерпретации и удаления заголовков нижних уровней по мере продвижения блока данных снизу вверх по уровням в компьютере. Этот процесс на каждом уровне распаковывает модуль передачи данных для вышестоящего уровня.

Десятичное представление с разделительными точками (Dotted-Decimal Notation — DDN). Формат IP-адресов версии 4, использующий четыре десятичных значения, разделенных точками.

Дистанционно-векторная (distance vector). Логика поведения некоторых внутренних протоколов маршрутизации, таких как RIP. Алгоритмы дистанционно-векторной маршрутизации запрашивают у каждого соседнего маршрутизатора всю его таблицу маршрутизации при каждом обновлении. Алгоритмы дистанционно-векторной маршрутизации могут быть склонны к

циклическим маршрутам, но в отношении вычислений они проще, чем алгоритмы маршрутизации, по состоянию канала.

Длина префикса (prefix length). Количество битов в префиксе IPv6.

Домен коллизий (collision domain). В сетях Ethernet область сети, в которой распространяются столкнувшиеся и поврежденные фреймы. Повторители и концентраторы не отфильтровывают такие поврежденные фреймы, в то время как коммутаторы локальных сетей LAN, мосты и маршрутизаторы их не пропускают.

Дуплексная передача (full-duplex). Возможность одновременной передачи данных между отправляющей и принимающей станциями в двух направлениях.

Enable-режим (enable mode). Привилегированный режим доступа к интерфейсу операционной системы Cisco IOS, в котором пользователь может вводить наиболее сложные и "опасные" для маршрутизатора или коммутатора команды, а также выполнять настройку устройства.

Заголовок (header). Набор байтов, помещаемый перед некими другими данными при инкапсуляции и определяемый в соответствии со специфическим протоколом.

Загрузочное поле (boot field). Четыре младших бита конфигурационного регистра маршрутизатора Cisco. Значение в загрузочном поле указывает маршрутизатору, из какого источника загружать операционную систему Cisco IOS.

Запись CIDR (CIDR notation). *См.* префиксная запись.

Запрос на получение информации о наличии маршрутизатора (Router Solicitation — RS). Сообщение, определенное протоколом обнаружения соседних устройств IPv6 (NDP), используемое для опроса всех маршрутизаторов о каналах связи, идентификаторах маршрутизатора и других параметрах конфигурации (префиксе и длине префикса).

Запрос о комментариях (Request For Comments — RFC). Серия документов IETF с описаниями набора протоколов Интернета и дополнительной информацией. Некоторые документы RFC приняты Советом по архитектуре Интернета (Internet Architecture Board — IAB) как стандарты Интернета. Большинство документов RFC определяют такие протоколы, как Telnet и FTP, но некоторые носят юмористический или исторический характер. Документы RFC доступны на многих веб-сайтах.

Запрос соседа (Neighbor Solicitation — NS). Сообщение, определенное протоколом обнаружения соседних устройств IPv6 (NDP), используемое для запроса у соседа анонса, содержащего список его MAC-адресов.

Зарезервированный порт (well-known port). Определены документом RFC 1700, зарезервированы и в протоколе TCP, и в протоколе UDP. Зарезервированные порты могут определять приложения, выполняемые над протоколами транспортного уровня.

Защита порта (port security). Коммутатор Cisco способен просматривать поступающие на интерфейс (порт) фреймы Ethernet, отслеживать MAC-адреса отправителей всех фреймов и при необходимости предпринимает действия по защите.

Звездообразная топология (star topology). Наиболее часто используемая физическая топология локальных сетей Ethernet. Сеть со звездообразной топологией имеет центральную точку соединений, которая может быть концентратором, коммутатором или маршрутизатором; в этой точке сходятся все кабельные сегменты.

Идентификатор маршрутизатора (Router ID — RID). 32-битовое число в протоколе OSPF, уникальным образом идентифицирующее каждый маршрутизатор.

Идентификатор подсети (IPv4) (subnet ID (IPv4)). *См.* номер подсети.

Идентификатор подсети (IPv6) (subnet ID (IPv6)). Число, представляющее подсеть IPv6. Оно же *префикс IPv6* (IPv6 prefix) или, более формально, *альтернативный адрес маршрутизатора подсети* (subnet router anycast address).

Изоляция проблемы (problem isolation). Этап процесса поиска и устранения неисправностей, на котором сетевой инженер пытается выделить основные причины отказа.

Именованный список управления доступом (named access list). Список ACL, в котором правила проверки пакетов идентифицируются на основании имени, а не номера.

Имя хоста (host name). Буквенно-цифровое обозначение хоста IP.

Инженерная группа по развитию Интернета (Internet Engineering Task Force — IETF). Группа IETF — это первичная организация, непосредственно создающая новые стандарты TCP/IP.

Инкапсуляция (encapsulation). Упаковка данных в заголовок некоторого конкретного протокола. Например, данные протоколов высокого уровня перед передачей помещаются в заголовок Ethernet. Аналогичным образом при мостовом соединении разнородных сетей весь фрейм из одной сети может быть помещен после заголовка, используемого протоколом канального уровня другой сети.

Институт инженеров по электротехнике и электронике (Institute of Electrical and Electronic Engineers — IEEE). Профессиональная организация, которая занимается разработкой коммуникационных и сетевых стандартов. Стандарты для сетей LAN, разработанные IEEE, в настоящее время преобладают при проектировании и эксплуатации сетей.

Интерфейс VLAN (VLAN interface). Концепция настройки коммутаторов Cisco, используемая как интерфейс между выполняющейся на коммутаторе операционной системой IOS, и поддерживаемыми им сетями VLAN, чтобы коммутатор мог присваивать им IP-адреса и посылать на них пакеты IP.

Интерфейс доступа (access interface). Термин, относящийся к дизайну локальных сетей; описывает интерфейс коммутатора, к которому подключены устройства конечных пользователей и настроенный так, чтобы не использовать магистральное соединение VLAN

Интерфейс командной строки (Command-Line Interface — CLI). Интерфейс, который позволяет пользователям взаимодействовать с операционной системой за счет ввода специализированных команд и их аргументов.

Источник синхроимпульсов (clock source). Устройство, с которым другие устройства в линии синхронизируют свою скорость в синхронных линиях.

Исходящий интерфейс (outgoing interface). Части записи таблицы маршрутизации маршрутизатора IP, относящаяся к локальному интерфейсу, на который локальный маршрутизатор должен перенаправить пакеты, соответствующие данному маршруту.

Исчерпание IPv4-адресов (IPv4 address exhaustion). Ожидается, что доступные для Интернета открытые IPv4-адреса, назначаемые с 1980-х годов, будут в конечном счете исчерпаны.

Кабельный Интернет (cable internet). Технология доступа к Интернету, использующая телевизионный кабель для передачи как видео, так и данных.

Канал связи (access link). В технологии Frame Relay — физический последовательный канал, соединяющий устройство DTE среды Frame Relay, обычно маршрутизатор, с коммутатором Frame Relay. В таком канале используются те же стандарты физического уровня, что и в двухточечных выделенных линиях.

Канал связи Ethernet (Ethernet link). Общее название любого физического канала связи между двумя узлами Ethernet, независимо от используемого кабельная.

Квартет (quartet). Термин, используемый только в этой книге, для обозначения набора из четырех шестнадцатеричных цифр в IPv6-адресе.

Классовая сеть IP (classful IP network). Сеть класса A, B или C протокола IPv4. Свое название получила потому, что подчиняется правилам классовой адресации.

Классовый протокол маршрутизации (classful routing protocol). Протокол, не передающий маску подсети в обновлениях совместно с адресом подсети и, следовательно, ориентирующийся на класс сети A, B или C. Не поддерживает маски VLSM.

Клиент DHCP (DHCP Client). Любое устройство, использующее протоколы DHCP для запроса в пользование IP-адреса от сервера DHCP и получения сервера любых других параметров IP.

Клиент WLAN (WLAN client). Беспроводное устройство, пытающееся подключиться к беспроводной сети через точку доступа.

Клиентский режим VTP (VTP client mode). Один из трех режимов работы протокола VTP на коммутаторах, при которых они узнают о номерах и именах сетей VLAN от других коммутаторов, но который не позволяет непосредственную настройку коммутатора.

Кодек (codec) — сокращение от *кодер-декодер* (coder-decoder). Устройство в виде интегральной схемы для преобразования аналоговых сигналов в цифровой поток битов и обратного преоб-

разования цифровых сигналов в аналоговые сигналы, обычно с помощью кодово-импульсной модуляции.

Коммутатор (switch). Устройство, соединяющее сегменты локальной сети LAN и использующее таблицу MAC-адресов для определения сегментов, в которые следует переслать фреймы. Такой принцип работы позволяет существенно уменьшить объем нецелесообразно рассылаемых данных. Коммутаторы работают с гораздо большими скоростями, чем мосты. Коммутаторы работают на канальном уровне эталонной модели OSI.

Коммутация без буферизации пакетов (cut-through switching). Устройства, использующие этот метод коммутации, читают, обрабатывают и пересылают фреймы сразу после того, как будет прочитан адрес получателя и определен порт назначения.

Коммутация каналов (circuit switching). Технология, в которой во время сеанса связи должен существовать физический канал между отправителем и получателем. Широко используется в сетях телефонных компаний. С технологической точки зрения коммутацию каналов можно рассматривать как противоположность коммутации пакетов и сообщений, а с точки зрения методов доступа — как противоположность методу конкуренции и передачи маркеров. Примером сетевой технологии с коммутацией каналов является ISDN.

Коммутация пакетов (packet switching). Сетевая технология, в которой разные хосты обмениваются пакетами данных по одному разделяемому каналу связи.

Коммутируемая сеть Ethernet (switched Ethernet). Сеть Ethernet, в которой используется коммутатор, но не концентратор, поэтому устройства в сети не конкурируют за среду передачи данных и ее полосу пропускания. Этот термин является антонимом термина *разделяемая сеть Ethernet*, который предполагает использование сети в режиме, в котором устройства конкурируют за среду передачи и полосу пропускания. Коммутируемый вариант сети позволяет наиболее оптимально использовать полосу пропускания и фактически выделить свою собственную полосу каждому устройству.

Коммутируемая телефонная сеть общего пользования (Public Switched Telephone Network — PSTN). Общее обозначение телефонных сетей и служб, в отличие от глобальных компьютерных сетей. Иногда используется аббревиатура POTS.

Конвергенция (convergence). Способность группы устройств объединенной сети, использующих конкретный протокол маршрутизации, согласовать друг с другом информацию о топологии сети после того, как в ней произошли изменения. Требуемое для этого время определяет скорость конвергенции.

Консольный порт (console port). Физический разъем в маршрутизаторе или коммутаторе, к которому кабелем может быть подключен компьютер. Этот порт используется для доступа к интерфейсу командной строки устройства и его конфигурированию, проверке с помощью программ эмуляции терминала.

Контрольная сумма фрейма (Frame Check Sequence — FCS). Поле в большинстве концевиков канала связи, используемое в процессе обнаружения ошибок.

Конфигурационный регистр (configuration register). 16-битовый конфигурируемый параметр в маршрутизаторах компании Cisco, который определяет, как работает маршрутизатор в процессе инициализации. Значение регистра устанавливается программными средствами в шестнадцатеричной системе с помощью специализированных команд.

Концевик (trailer). Набор байтов, помещаемый позади неких других данных и инкапсулирующий эти данные так, как определено соответствующим протоколом. Как правило, концевики определяют только протоколы канального уровня.

Концентратор (hub). Общая точка соединений устройств сети. Обычно концентраторы используются для подключения отдельных сегментов к локальной сети. Концентратор может иметь несколько портов. Когда на один из них поступает пакет, он копируется и направляется на все остальные порты концентратора, поэтому такой пакет поступает во все сегменты сети LAN.

Лавинная рассылка (flooding). Способ передачи данных, применяемый коммутаторами и мостами, при использовании которого данные, полученные на некотором интерфейсе, рассылаются

через все интерфейсы устройства, за исключением того, на котором они были первоначально получены.

Логический адрес (logical address). Общий термин, обозначающий адреса сетевого уровня и протоколов третьего уровня. Он не зависит от нижних уровней и зачастую противопоставляется адресам канального уровня, называемым физическими, поскольку последние зависят от используемой технологии физического уровня.

Локальная область видимости канала связи (link-local scope). При групповой передаче IPv6 обозначает часть (область видимости) сети, в которую может быть передан многоадресатный пакет. Термин "локальный в пределах канала связи" отражает тот факт, что пакет остается в той подсети, в которую он попал.

Локальное имя пользователя (local username). Имя пользователя (с паролем), настроенное на маршрутизаторе или коммутаторе. Оно считается локальным, поскольку существует на маршрутизаторе или коммутаторе, а не на дистанционном сервере.

Магистральное соединение (trunking), или магистральное соединение VLAN. Метод, использующий протоколы Cisco ISL или IEEE 802.1Q, для поддержки нескольких соединений VLAN, позволяя их трафику течь по одному каналу связи.

Магистральный интерфейс (trunk interface). Физическое и логическое соединение между двумя коммутаторами, по которому передается сетевой трафик. В современных коммутаторах используются магистральные соединения стандартов 802.1Q или ISL.

Магистральный канал (trunk). Сегмент территориальной локальной сети Ethernet, на котором устройства добавляют во фрейм заголовок, идентифицирующий сеть VLAN.

Маршрут хоста (host route). Маршрут с маской /32, представляющий собой маршрут к одному IP-адресу хоста.

Маршрутизируемый протокол (routed protocol). Протокол, отвечающий за передачу данных, например IPv4 или IPv6.

Маска подсети (subnet mask). 32-разрядная маска адреса, используемая протоколом IP для описания битов части сети и хоста в адресе подсети. Части сети адреса соответствуют двоичные 1 в маске, а части хоста — 0.

Маска подсети переменной длины (Variable-Length Subnet Mask — VLSM). Возможность задавать различные маски для одной и той же сети класса A, B или C в различных подсетях. Маска VLSM позволяет оптимизировать доступное адресное пространство.

Международная организация по стандартизации (International Organization for Standardization — ISO). Международная ассоциация национальных организаций по стандартизации, обеспечивающая разработку и поддержку глобальных стандартов в сфере коммуникаций и обмена информацией. ISO разработала популярную эталонную модель взаимодействия открытых систем — OSI.

Межкоммутаторный канал (Inter-Switch Link — ISL). Собственный протокол компании Cisco, который сохраняет информацию виртуальной сети LAN при обмене трафиком между коммутаторами и маршрутизаторами.

Межсетевая операционная система корпорации Cisco (Internetwork Operating System — Cisco IOS). Программное обеспечение межсетевой операционной системы корпорации Cisco обеспечивает функциональность, расширяемость и безопасность всех аппаратных продуктов. Программное обеспечение хранится в виде образа во флеш-памяти, загружается в оперативную память устройства, обеспечивает его работу и выполнение всех необходимых функций.

Метод скользящего окна (sliding window). Метод управления потоком, при котором получатель дает отправителю разрешение на пересылку данных до заполнения окна. Когда окно заполняется, отправитель прекращает передачу до тех пор, пока получатель не объявит о расширении окна. Этот метод управления потоком используется в протоколе TCP и других транспортных протоколах, а также в некоторых протоколах канального уровня.

Метрика (metric). Числовое значение, вырабатываемое каким-либо алгоритмом для каждого маршрута в сети. Обычно чем меньше метрика, тем предпочтительнее маршрут.

Механизм создания подсетей (subnetting). Метод деления полного адреса любого из классов на более мелкие части. Этот механизм позволил избежать полного исчерпания доступных IP-адресов (версии 4).

Микросегментация (microsegmentation). Позволяет создавать в локальной сети частные или выделенные сегменты, в которых на каждый сегмент приходится только одна рабочая станция. В этом случае каждая станция получает мгновенный доступ ко всей полосе пропускания, и ей не приходится конкурировать с другими за доступ к имеющейся полосе пропускания.

Многоадресатный адрес всех маршрутизаторов (all-routers multicast address). Специфический многоадресатный IPv6-адрес, FF02::2, с локальной областью видимости канала связи и используемый для передачи пакетов на все устройства, являющиеся маршрутизаторами IPv6 в локальном канале связи.

Многоадресатный адрес всех узлов (all-nodes multicast address). Специфический многоадресатный IPv6-адрес, FF02::1, с локальной областью видимости канала связи и используемый для передачи пакетов на все устройства канала связи, поддерживающие протокол IPv6.

Многоадресатный адрес опрашиваемого узла (solicited-node multicast address). Тип многоадресатного адреса IPv6 с локальной областью видимости канала связи, используемый для передачи пакетов всем хостам в подсети, имеющим одинаковое значение в последних шести шестнадцатеричных цифрах своих одноадресатных IPv6-адресов. Начинается с FF02::1:FF00:0/104.

Многомодовый оптоволоконный кабель (Multimode Fiber — MM fiber). Оптоволоконный носитель, в котором свет распространяется в нескольких модах за счет того, что диаметр его больше и пучок света может входить под разными углами. Полоса пропускания такого кабеля меньше, чем у одномодового, но в нем обычно используются более дешевые источники световых импульсов — светодиодные излучатели, а не лазеры.

Многоуровневый коммутатор (multilayer switch). Коммутатор LAN, способный также выполнять функции маршрутизации уровня 3. Название свидетельствует о том, что это устройство принимает решения о перенаправлении на основании логики нескольких уровней OSI (уровней 2 и 3).

Множественный доступ с предотвращением коллизий (Carrier Sense Multiple Access/Collision Avoidance — CSMA/CA). Механизм доступа к среде передачи, при использовании которого устройства стараются избежать коллизий за счет использования специального фрейма. Такой метод доступа к среде используется в беспроводных сетях.

Модем (modem) — сокращение от *модулятор-демодулятор* (modulator-demodulator). Устройство, преобразующее цифровые сигналы в аналоговые, и наоборот. На станции-отправителе модем преобразует цифровые сигналы в форму, соответствующую каналам аналоговой связи. В пункте назначения аналоговые сигналы преобразуются в цифровую форму. Модемы позволяют передавать информацию по обычным телефонным линиям.

Модем DSL (DSL modem). Устройство, использующее стандарты DSL, для обмена данными по телефонной линии с телефонной компанией, использующее тот же стандарт.

Модуль обслуживания канала/модуль обработки данных (Channel Service Unit/Data Service Unit — CSU/DSU). Устройство, работающее со стандартами первого уровня в последовательных каналах, устанавливаемое оператором связи и определяющее, как телекоммуникационное оборудование будет взаимодействовать с последовательным портом маршрутизатора.

Направленный широковещательный адрес (directed broadcast address). *См.* широковещательный адрес подсети.

Неполносвязная топология (partial-mesh topology). В сети с такой топологией, по крайней мере, одно устройство имеет несколько соединений с другими устройствами сети, однако при этом сеть не обладает полносвязной структурой. Вместе с тем неполносвязная топология обеспечивает определенный уровень избыточности за счет наличия нескольких альтернативных маршрутов.

Неэкранированная витая пара (Unshielded Twisted Pair — UTP). Среда передачи данных, представляющая собой четыре пары свитых медных проводников в пластиковой оболочке без металлизированного экрана. Широко используется в различных сетях.

Номер порта (port number). Поле в заголовке TCP или UDP, идентифицирующее приложение, которое пересылает (порт отправителя) или принимает (порт получателя) поток сегментов данных.

Нулевая подсеть (zero subnet). Подсеть в классовой сети IPv4, в которой в адресе подсети стоят двоичные нули. В десятичном виде нулевая подсеть может быть легко идентифицирована, поскольку ее адрес совпадает с адресом сети.

Обнаружение ошибок (error detection). Процесс определения, был ли фрейм канального уровня изменен в процессе передачи. Для решения этой задачи обычно используется *контрольная сумма фрейма* (Frame Check Sequence — FCS) в концевике фрейма.

Оборудование клиента (Customer Premises Equipment — CPE). Оконечное оборудование (терминалы, телефоны, модемы и т.п.), устанавливаемое телефонной компанией у клиента и подключенное к сети телефонной компании.

Образ IOS (IOS Image). Файл, содержащий операционную систему IOS.

Общественная телефонно-телеграфная компания (Post, Telephone and Telegraph — PTT). Коммерческая компания или правительственная организация, обеспечивающая телефонное обслуживание. Отделения PTT существуют во всех странах и обеспечивают местную и междугородную телефонную связь.

Одноадресатный IP-адрес (unicast IP address). IP-адрес, представляющий один интерфейс. В протоколе IPv4 эти адреса относятся к диапазонам классов A, B и C.

Одноадресатный адрес (unicast address). Любой сетевой адрес, представляющий одно устройство или интерфейс, а не группу адресов (ее представляют многоадресатный и широковещательный адреса).

Одноадресатный фрейм с неизвестным получателем (unknown unicast frame). Фрейм Ethernet, MAC-адрес получателя которого отсутствует в таблице коммутации (таблице MAC-адресов) коммутатора, поэтому устройство должно разослать его с использованием лавинного механизма.

Одномодовый оптоволоконный кабель (single-mode fiber). Разновидность оптического волокна с тонким сердечником, в который луч света может входить под одним определенным углом. Пропускная способность такого волокна много больше, чем у многомодового, но для передачи данных нужен источник с узким спектром сигнала, например лазер.

Окно (window). Количество битов, которые могут быть пересланы без подтверждения.

Оперативная память (Random-Access Memory — RAM). Память, которая функционирует только при включенном питании, ее содержимое может записываться и считываться микропроцессором.

Определение дублирующегося адреса (Duplicate Address Detection — DAD). Согласно протоколу IPv6, хост сначала проверяет, не использует ли другой хост тот адрес, который он хочет использовать.

Отказ в обслуживании (Denial of Service — DoS). Тип сетевой атаки, призванной заблокировать сеть, завалив ее потоком бесполезного трафика.

Открытый IP-адрес (public IP address). Адрес, являющийся частью зарегистрированной сети или диапазона сетей регионального агентства *Центра по присвоению адресов Интернета* (Internet Assigned Numbers Authority — IANA). Такой адрес может использовать только организация, зарегистрировавшая его на себя, и маршрутизаторы в Интернете будут знать маршруты к открытым зарегистрированным IP-адресам.

Открытый протокол поиска первого кратчайшего маршрута (Open Shortest Path First — OSPF). Популярный протокол маршрутизации внутреннего шлюза (IGP), использующий для вычисления наилучших маршрутов достижения каждой известной подсети базу данных состояния каналов и алгоритм поиска кратчайших маршрутов (SPF).

Пакет (packet). Логически сгруппированная информация, состоящая из заголовка, содержащего управляющую информацию, и (как правило) данных пользователя. Чаще всего пакетами называют блоки данных сетевого уровня. На разных уровнях эталонной модели OSI и в разных областях техники для описания логического группирования информации используются термины *дейтаграмма*, *фрейм*, *сообщение* и *сегмент*.

Пакет IP (IP packet). Состоит из заголовка IP, сопровождаемого инкапсулируемыми данными, но не включает любые заголовки и концевики для уровней ниже сетевого.

Первопричина (root cause). Стандартный термин в поиске и устранении неисправностей, описывающий причину неработоспособности чего-либо. Основная задача сетевого инженера — найти и устранить такую причину, в результате проблема будет или решена, или перейдет в другую проблему.

Передача голоса по сети IP (Voice over IP — VoIP). Позволяет маршрутизатору передавать по объединенным сетям IP голосовой трафик.

Перекрещенный кабель (crossover cable). Кабель, в котором перекрещена пара, чтобы правильно подать, передать и получить сигналы между однотипными устройствами. В сетях 10BASE-T и 100BASE-TX провода контактов 1 и 2 подключены к контактам 3 и 6 на другом конце кабеля, а 3 и 6, соответственно, ведут к контактам 1 и 2 на другом конце кабеля.

Перекрывающиеся подсети (overlapping subnets). Результат неправильного дизайна подсетей IP, когда диапазон адресов одной подсети перекрывается диапазоном другой.

Пересылка данных (forwarding). Процесс передачи фрейма к получателю с одного интерфейса устройства на другой.

Плата сетевого интерфейса (Network Interface Card — NIC). Компьютерная плата расширения или интегрированная часть системной платы, обеспечивающая подключение к компьютерной сети. Ныне большинство сетевых плат являются платами Ethernet, обладающих портом RJ-45, наиболее распространенным типом порта Ethernet.

Побитовое булево "И" (bitwise Boolean AND). Операция логического "И" для двух чисел одинаковой длины, выполняемая сначала для первого бита числа, потом для второго, для третьего и т.д.

Подключенный маршрут (connected route). Маршрут IP, добавленный в таблицу маршрутизации маршрутизатора, когда интерфейс маршрутизатора подключен и имеет IP-адрес. Маршрут для подсети, которая может быть вычислена на основании заданного IP-адреса и маски.

Подсеть (subnet). Некоторая часть сети класса A, B или C, выделенная сетевым инженером. С помощью подсетей можно сэкономить адресное пространство и создать группы IP-адресов.

Подсеть IP (IP subnet). Подразделение сети класса A, B или C, созданное сетевым администратором. Подсети позволяют использовать одну сеть класса A, B или C вместо нескольких сетей.

Подуровень управления доступом к передающей среде (Media Access Control — MAC). Нижний из двух подуровней канального уровня, определенных спецификацией IEEE. Подуровень MAC управляет доступом к передающей среде, таким как передача маркера или конкуренция за доступ.

Подуровень управления логическим каналом (Logical Link Control — LLC). Верхний из двух подуровней канального уровня, определенных в стандарте IEEE. Подуровень LLC осуществляет контроль ошибок, управление потоками, создание фреймов и адресацию на уровне MAC. Основным протоколом LLC является спецификация IEEE 802.2, которая описывает как вариант сети с установкой соединения, так и без него.

Поле типа протокола (protocol type field). Поле в заголовке фрейма в локальной сети, идентифицирующее следующий за ним заголовок верхнего уровня. Включает в себя поле DIX Ethernet Type, поле стандарта IEEE 802.2 DSAP и поле типа протокола SNAP.

Полносвязная топология (full mesh). Вариант топологии сети, в которой все устройства соединены со всеми и могут взаимодействовать напрямую.

Полудуплексная передача (half duplex). Возможность передачи данных между передающей и принимающей станциями в каждый конкретный момент времени только в одном направлении.

Пользовательский режим (user mode). Режим интерфейса командной строки маршрутизатора или коммутатора, в котором пользователь может вводить команды, не влияющие на работу устройства, обычно просматривать текущее состояние служб и портов, но не может настраивать устройство.

Порт (port). В терминологии IP — процесс верхнего уровня, который принимает данные от нижних уровней. Порты нумеруются и привязываются к конкретным процессам. Например, про-

токол SNMP приписан к порту с номером 25. Номер порта такого типа называется *зарезервированным портом*, или *адресом*.

Порт Ethernet (Ethernet port). Общее название любых гнезд на обратной стороне любого устройства Ethernet, как правило, сетевой платы Ethernet или коммутатора LAN, в который может быть вставлен разъем кабеля Ethernet.

Последовательный интерфейс (serial interface). Тип интерфейса маршрутизатора, используемый для подключения некоторых типов WAN, в частности выделенных линий и каналов связи Frame Relay.

Последовательный кабель (serial cable). Разновидность кабеля, в котором используются разные интерфейсы для подключения к модулю или устройству CSU/DSU выделенной линии.

Постоянное запоминающее устройство (Read-Only Memory — ROM). Тип компьютерной памяти, в которой данные записаны предварительно.

Префикс (prefix). Число, идентифицирующее группу IPv6-адресов. Идентификатор подсети IPv6.

Префиксная запись (prefix notation) (IP версии 4). Краткий формат записи маски подсети, в котором указывается только количество единичных (содержащих 1) битов в маске после косой черты. Например, /24 описывает маску с 24 единичными битами, т.е. маску сети класса C. Зачастую количество единичных битов в маске сети называют длиной префикса.

Проверка доступности адресата (Packet Internet Groper — Ping). Команда, используемая для отправки эхо-запроса протокола ICMP и получения на него ответа. Часто используется в сетях IP для проверки наличия связи с сетевым устройством.

Прозрачный мост (transparent bridge). Устройство, которое было предшественником современных коммутаторов локальных сетей. Мосты пересылают фреймы между сегментами локальной сети на основании MAC-адреса получателя, как и коммутаторы, а прозрачными их называют потому, что их присутствие в сети незаметно для оконечных устройств.

Прозрачный режим VTP (VTP transparent mode). Один из трех режимов работы протокола VTP, допускающий настройку VLAN коммутатора, но не позволяющий ему информировать об изменениях VLAN другие коммутаторы и узнавать от них о таких изменениях.

Протокол EIGRP версии 6 (EIGRP version 6). Версия протокола маршрутизации EIGRP, поддерживающая протокол IPv6, а не IPv4.

Протокол Frame Relay. Стандартный протокол коммутируемой передачи данных канального уровня, который управляет несколькими виртуальными каналами между подключенными устройствами с помощью инкапсуляции HDLC. Он эффективнее протокола X.25 и обычно рассматривается как его замена.

Протокол HTTP. Протокол передачи гипертекста (Hypertext Transfer Protocol), используемый веб-браузерами и веб-серверами для передачи файлов, например текстовых и графических.

Протокол IP версии 4 (IP version 4). Версия протокола Интернета, определенная документом RFC 791, стандартизированная в 1980 году и используемая как основа сетей TCP/IP и Интернета больше 30 лет.

Протокол IP версии 6 (IP version 6). Более новая версия протокола Интернета, определенная документом RFC 2460, наравне с множеством запросов на комментарии, созданные во избежание проблемы исчерпания IPv4-адресов.

Протокол OSPF версии 2 (OSPF version 2). Версия протокола маршрутизации OSPF, поддерживающая протокол IPv4, а не IPv6. Была общепринята 20 лет назад.

Протокол OSPF версии 3 (OSPF version 3). Версия протокола маршрутизации OSPF, поддерживающая протокол IPv6, а не IPv4.

Протокол Secure Shell (SSH). Протокол уровня приложений TCP/IP, позволяющий эмулировать терминальное подключение к серверу и использующий динамический обмен ключами и шифрование, для обеспечения безопасности соединения.

Протокол двухточечного соединения (Point-to-Point Protocol — PPP). Преемник протокола SLIP, который обеспечивает соединения "маршрутизатор–маршрутизатор" и "хост–сеть" по синхронным и асинхронным каналам. Протокол SLIP был разработан для работы с IP, но PPP может работать с несколькими протоколами сетевого уровня, такими как IP, IPX и ARA.

Протокол динамического конфигурирования хоста (Dynamic Host Configuration Protocol — DHCP). Обеспечивает механизм динамического распределения и повторного использования освобождаемых IP-адресов, а также автоматическую настройку маски, стандартного шлюза и IP-адреса сервера DNS.

Протокол Интернета (Internet Protocol — IP). Протокол сетевого уровня в стеке протоколов TCP/IP; обеспечивает передачу данных между сетями без предварительной установки соединения.

Протокол маршрутизации (routing protocol). Протокол, осуществляющий реализацию какого-либо алгоритма маршрутизации. Примерами протоколов маршрутизации могут служить протоколы IGRP, OSPF и RIP.

Протокол маршрутизации внешнего шлюза (Exterior Gateway Protocol — EGP). Протокол Интернета, использующийся для обмена маршрутной информацией между автономными системами.

Протокол маршрутизации внутреннего шлюза (Interior Gateway Protocol — IGP). Протокол Интернета, использующийся для обмена маршрутной информацией в автономных системах. Примерами широко используемых протоколов класса IGP являются IGRP, OSPF и RIP.

Протокол обнаружения соседних устройств (Neighbor Discovery Protocol — NDP). Часть семейства протоколов IPv6, используемая для обнаружения информации о соседних устройствах и обмена информацией о них в той же подсети. Заменяет протокол ARP технологии IPv4.

Протокол обнаружения устройств Cisco (Cisco Discovery Protocol — CDP). Протокол CDP используется для получения информации о соседних устройствах, такой, как тип присоединенных устройств, интерфейсы маршрутизатора, которые в настоящий момент присоединены, и номера моделей устройств. Данный протокол работает практически во всем оборудовании компании Cisco: в коммутаторах, маршрутизаторах, серверах доступа и др.

Протокол передачи пользовательских дейтаграмм (User Datagram Protocol — UDP). Является протоколом транспортного уровня без установления соединения из группы протоколов TCP/IP. UDP — это простой протокол, который обеспечивает обмен дейтаграммами без подтверждений или гарантий доставки, требуя, чтобы обработку ошибок и повторную передачу контролировал какой-либо другой протокол. Этот протокол описан в документе RFC 768.

Протокол преобразования адресов (Address Resolution Protocol — ARP). Протокол Интернета, используемый для преобразования IP-адресов в MAC-адреса. Определен документом RFC 826.

Протокол распределенного связующего дерева (Spanning Tree Protocol — STP). Используемый в мостах и коммутаторах протокол, в котором задействован алгоритм связующего дерева для обеспечения динамического самообучения мостов и предотвращения образования петлевых маршрутов. Мосты обмениваются сообщениями BPDU, которые позволяют обнаружить петлевые маршруты и устранить их, отключая отдельные интерфейсы.

Протокол синхронизации сетевого времени (Network Time Protocol — NTP). Позволяет нескольким устройствам использовать одинаковое время дня, что облегчает поиск сообщений в журналах на основании временных меток.

Протокол создания магистралей VLAN (VLAN Trunking Protocol — VTP). Собственный протокол компании Cisco для обмена информацией о конфигурации между коммутаторами Cisco, включая данные о существующих сетях VLAN, их идентификаторах и именах.

Протокол третьего уровня (layer 3 protocol). Протокол, соответствующий требованиям уровня 3 эталонной модели OSI, определяющий принципы маршрутизации и адресации в сети. Примерами протоколов третьего уровня являются IPv4 и IPv6.

Протокол управления передачей (Transmission Control Protocol — TCP). Протокол транспортного уровня с установлением соединения, который обеспечивает надежную дуплексную передачу. Относится к стеку TCP/IP.

Протокол управления передачей/протокол Интернета (Transmission Control Protocol/Internet Protocol — TCP/IP). Название набора протоколов, разработанных Министерством обороны США в 1970-х годах для создания всемирной сети. TCP и IP — два наиболее известных протокола из этого стека.

Протокол управляющих сообщений Интернета (Internet Control Message Protocol — ICMP). Представляет собой протокол Интернета сетевого уровня стека TCP/IP, сообщающий об ошибках и предоставляющий другую информацию относительно обработки пакетов IP. Описан в документе RFC 792.

Прямое лабораторное подключение (back-to-back link). Последовательный канал между двумя маршрутизаторами без модулей CSU/DSU, установленный за счет подключения кабеля DTE одного маршрутизатора к кабелю DCE другого. Обычно такое подключение используется в лабораторных работах для соединения двух устройств без использования выделенной линии от местного оператора связи.

Прямое подтверждение (forward acknowledgment). Процесс, используемый протоколами, осуществляющими восстановление после ошибок, при котором подтверждающее данные число перечисляет следующие данные, подлежащие передаче, а не последние успешно полученные данные.

Прямой кабель (straight-through cable). Кабель, в котором сохранен порядок следования контактов на обоих концах. Если провод подсоединен к контакту с номером 1 на одном конце кабеля, то и на другом конце он будет подсоединен к аналогичному контакту 1.

Рабочий режим магистрали (trunking operational mode). Стандартный режим интерфейса коммутатора Cisco для магистрального соединения VLAN.

Распределенная сеть (Wide-Area Network — WAN). Часть большей сети, реализующая технологии уровня 1 и 2 модели OSI. Соединяет площадки, обычно находящиеся далеко друг от друга, и использует бизнес-модель, подразумевающую аренду потребителем (человеком или предприятием) сети WAN у провайдера услуг (зачастую телефонной компании).

Рассогласование дуплекса (duplex mismatch). Состояние, при котором устройства на двух противоположных концах любого канала связи Ethernet используют разную логику дуплексной передачи (один полный дуплекс, второй полудуплекс), приводящее к порче фреймов и повторной передаче на канале связи.

Расширение спектра со скачкообразным изменением частоты (Frequency Hopping Spread Spectrum — FHSS). Метод кодирования данных в беспроводной сети, в котором передача каждого следующего блока данных осуществляется на близко расположенной, но другой частоте. Не используется в современных стандартах беспроводных сетей.

Расширенный протокол маршрутизации внутреннего шлюза (Enhanced Interior Gateway Routing Protocol — EIGRP). Обновленная версия протокола IGRP, разработанная компанией Cisco. Для этого протокола характерна очень быстрая конвергенция, высокая эффективность и экономия полосы пропускания сети. Сочетает в себе самые лучшие функции дистанционно-векторных протоколов и протоколов с учетом состояния каналов.

Расширенный список управления доступом (extended access list). Набор команд `access-list` в глобальной конфигурации системы IOS, при помощи которого проверяются различные характеристики пакета IP, например IP-адрес отправителя и получателя, а также порты TCP и UDP, чтобы принять решение о том, следует ли отбросить такой пакет или передать дальше.

Режим коммутации без фрагментации (fragment-free switching). Режим коммутации, при котором перед пересылкой фреймов фильтруются фрагменты коллизий, являющиеся основным источником ошибок в сети.

Режим коммутации с буферизацией пакетов (store-and-forward switching). Техника коммутации, при которой фрейм перед пересылкой в порт назначения полностью записывается в память устройства и обрабатывается. Обработка включает в себя подсчет контрольной суммы и проверку адреса получателя. Дополнительно фрейм должен быть временно сохранен до тех пор, пока сетевые ресурсы (например, канал) не станут доступны для пересылки фрейма.

Режим конфигурации (configuration mode). Используется для ввода однострочных команд и команд, которые вносят изменения в глобальную конфигурацию маршрутизатора. Команды сохраняются в текущем файле конфигурации устройства (`running-config`).

Режим начальной установки (setup mode). Окно операционной системы Cisco IOS в маршрутизаторах и коммутаторах, позволяющее настроить базовые параметры устройства в режиме интерактивного диалога и создать файл текущей и стартовой конфигурации.

Ретранслятор DHCP (DHCP Relay). Средство маршрутизатора IOS, передающее сообщения DHCP от клиента на серверы при изменении IP-адреса получателя с 255.255.255.255 на IP-адрес сервера DHCP.

Самообучение (learning). Процесс, используемый коммутаторами для обнаружения MAC-адресов и их относительного месторасположения. Коммутаторы просматривают MAC-адреса отправителей для всех входящих фреймов с целью построения таблицы коммутации.

Сбалансированный гибрид (balanced hybrid). Термин, использовавшийся в последние годы для обозначения логики протокола маршрутизации EIGRP. Сегодня эту логику обычно называют *улучшенной дистанционно-векторной* (advanced distance vector).

Сегмент (segment). 1. Часть сети, ограниченная мостами, маршрутизаторами или коммутаторами, например в сети Ethernet. В среде Ethernet сегмент может представлять собой как один участок кабеля, так и единый домен коллизий, в котором есть много кабелей. 2. В спецификации протокола TCP — логически сгруппированная информация на транспортном уровне эталонной модели OSI, иногда называемая *L4PDU*.

Сегментация (segmentation). Процесс разделения больших блоков данных от приложения на маленькие, размер которых соответствует используемой среде и технологии передачи данных.

Сервер DHCP (DHCP Server). Программное обеспечение, ожидающее от клиентов DHCP запросы на пользование IP-адресов и присваивающее резервируемые IP-адреса, а также сообщающее клиенту другие важные параметры IP.

Сервер DHCP без фиксации состояния (stateless DHCP). Термин протокола IPv6, описывающий ситуацию, когда сервер не выдает IPv6-адреса клиентским устройствам, но предоставляет другую вспомогательную информацию, например IP-адрес сервера DNS, и не отслеживает информацию о клиентах (информацию о состоянии).

Сервер DHCP с фиксацией состояния (stateful DHCP). Термин протокола IPv6, описывающий ситуацию, когда сервер отслеживает, каким именно клиентским устройствам какие IPv6-адреса были выданы (информация о состоянии).

Сервер имен (name server). Сервер, установленный в сети, где запущена служба преобразования сетевых имен в IP-адреса, и наоборот.

Серверный режим VTP (VTP server mode). Один из трех режимов работы протокола VTP, допускающий настройку VLAN коммутатора, позволяющий ему информировать об изменениях VLAN другие коммутаторы и узнавать от них о таких изменениях.

Сетевая модель (networking model). Общий термин, описывающий некоторый набор протоколов и стандартов, объединяемых по некоторому признаку. Впоследствии согласно модели разрабатываются сетевые устройства, позволяющие объединить хосты в сеть. Примерами сетевых моделей являются TCP/IP и OSI.

Сетевая часть адреса (network part). Часть адреса длиной 1, 2 или 3 октета (байта) в стандарте IPv4, основанная на классе сети A, B или C.

Сетевой адрес (network address, network number). Адрес сетевого уровня, который относится к логическому, а не физическому сетевому устройству. Его также называют *протокольным адресом* (protocol address).

Сеть (network). Группа компьютеров, принтеров, маршрутизаторов, коммутаторов и других устройств, которые могут взаимодействовать в некоторой среде передачи.

Сеть IP (IP network). *См.* классовая сеть IP.

Симметричность (symmetric). Характеристика многих технологий доступа к Интернету, предполагающая, что нисходящий и восходящий каналы имеют одинаковую скорость передачи данных и емкость.

Синхронизация (clocking). Процесс передачи специализированного служебного сигнала по кабелю в основной полосе пропускания или по отдельному контакту, по которому принимающее устройство согласует свою работу с передающим устройством.

Синхронная оптическая сеть (Synchronous Optical Network — SONET). Спецификация высокоскоростной (свыше 2,5 Гбит/с) синхронной сети на базе оптоволокна, разработанная Bellcore. Основным блоком сети SONET является STS-1. В 1988 году стандарт SONET был утвержден в качестве международного.

Синхронность (synchronous). Вставка временных меток в поток битов. На практике устройство на одном конце линии подстраивается под скорость передачи другого конца линии, тем не менее, принимая поток данных, оборудование обнаруживает небольшие отклонения и должно постоянно подстраивать свою скорость.

Система доменных имен (Domain Name System — DNS). Система, используемая в Интернете для трансляции имен хостов в сетевые адреса.

Система обнаружения вторжений (Intrusion Detection System — IDS). Средство безопасности в сетях, исследующее сложные шаблоны трафика и сравнивающее их с известными сигнатурами и профилями атак, позволяющее классифицировать атаки на сети и уведомлять сетевого администратора.

Система предотвращения вторжений (Intrusion Prevention System — IPS). Средство безопасности в сетях, исследующее сложные шаблоны трафика и сравнивающее их с известными сигнатурами и профилями атак, позволяющее классифицировать атаки и предпринимать определенные действия по их предотвращению.

Следующий транзитный маршрутизатор (next-hop router). Часть записи таблицы маршрутизации маршрутизатора IP, относящаяся к следующему маршрутизатору IP (по IP-адресу), который должен получить пакеты, соответствующие данному маршруту.

Служба telnet. Стандартный протокол эмуляции терминала из группы протоколов TCP/IP. Протокол telnet используется для организации подключений с дистанционного терминала и позволяет пользователям входить в дистанционную систему и использовать ее ресурсы так, словно они подключены к локальной системе. Описан в документе RFC 854.

Совместно используемая сеть Ethernet (shared Ethernet). Сеть Ethernet, в которой используется концентратор или коаксиальный кабель и устройства должны конкурировать за возможность передачи в такой среде, разделяя ее ресурсы между собой.

Соединительный кабель (patch cable). Кабель Ethernet, обычно короткий, соединяющий порт Ethernet устройства с розеткой на стене или коммутатором. При монтаже в здании электрики используют соединительные кабели для подключения кабельных узлов к розеткам в комнатах и других местах.

Сосед (neighbor). В протоколах маршрутизации другой маршрутизатор, с которым данный маршрутизатор решает обмениваться информацией о маршрутизации.

Сосед CDP (CDP neighbor). Устройство на другом конце некоторого телекоммуникационного кабеля, рассылающее пакеты CDP.

Состояние "up и up" (up and up). Жаргонное выражение, означающее два состояния интерфейса маршрутизатора или коммутатора Cisco IOS (состояние линии и состояние протокола). Первое "up" означает состояние линии, а второе — состояние протокола. Интерфейс в этом состоянии должен быть способен передавать фреймы канала связи.

Состояние канала (link state). Один из классов алгоритмов, используемых в протоколах маршрутизации. Протоколы с учетом состояния каналов строят базу данных со множеством деталей о каналах (подсетях) и их состоянии (работает, выключен), на основании которой строится таблица оптимальных маршрутов.

Спецификация Ethernet. Базовая спецификация LAN, созданная корпорацией Xerox и впоследствии развивавшаяся корпорациями Xerox, Intel и Digital Equipment, а затем утвержденная IEEE.

Спецификация IEEE 802.2. Протокол IEEE для локальных сетей LAN, определяющий реализацию подуровня LLC канального уровня эталонной модели OSI. Стандарт IEEE 802.2 задает методы обработки ошибок и интерфейс службы сетевого (третьего) уровня.

Спецификация IEEE 802.3. Протокол IEEE для сетей LAN, определяющий реализацию подуровня MAC канального уровня (т.е. физическую часть последнего). В спецификации IEEE 802.3 ис-

пользуется метод доступа CSMA/CD для набора возможных скоростей передачи данных в разнообразных физических средах.

Стандарт EUI-64. Стандарт расширенного уникального идентификатора (Extended Unique Identifier) длиной 64 бита.

Стандартная маска (default mask). Маска, используемая в сетях класса A, B или C, которая не создает каких-либо подсетей. Сети класса A соответствует маска 255.0.0.0, класса B — 255.255.0.0, класса C — 255.255.255.0.

Стандартный маршрут (default route). Маршрут в маршрутизаторе, по которому отправляются все пакеты, сеть получателя для которых отсутствует в явном виде в таблице маршрутизации.

Стандартный список доступа (standard access list). Список в конфигурации IOS, позволяющий проверить только IP-адрес отправителя пакета и по нему принять решение, передавать пакет дальше или отбросить его.

Стандартный шлюз/стандартный маршрутизатор (default gateway/default router). В конфигурации хоста IP — IP-адрес маршрутизатора, которому узел будет пересылать пакеты в том случае, если IP-адрес получателя пакета относится к другой подсети.

Файл стартовой конфигурации (`startup-config file`). В коммутаторах и маршрутизаторах компании Cisco под управлением операционной системы IOS так называется файл, размещаемый в памяти NVRAM устройства, в котором хранятся настройки устройства, используемые при загрузке и копируемые в текущий файл конфигурации в оперативной памяти в процессе запуска.

Статический маршрут (static route). Маршрут IP на маршрутизаторе, созданный и настроенный пользователем на локальном маршрутизаторе.

Схема расположения выводов (pinout). Схема подключения проводников в кабеле к контактам разъема.

Таблица ARP (ARP table). Хранимый в памяти хостов и маршрутизаторов список IP-адресов соседних устройств той же сети VLAN наряду с их MAC-адресами.

Таблица маршрутизации (routing table). Представляет собой некоторую разновидность базы данных, хранящуюся в маршрутизаторе или другом устройстве объединенной сети, в которой содержится информация о маршрутах к конкретным сетям-получателям и в большинстве случаев метрики, связанные с этими маршрутами.

Таблица соседних устройств IPv6 (IPv6 neighbor table). Эквивалент IPv6 таблицы ARP. Таблица содержит перечень IPv6-адресов других хостов на том же канале связи наряду с их MAC-адресами. Обычно составляется с использованием протокола обнаружения соседних устройств (NDP).

Таймер бездействия (inactivity timer). Параметр таблицы MAC-адресов коммутатора, который связан с каждой из записей таблицы и отсчитывается от нуля или сбрасывается до нуля в том случае, когда коммутатор получил фрейм с имеющимся у него в таблице MAC-адресом. Записи с наибольшими значениями таймера удаляются первыми в том случае, если нужно освободить место в таблице для новых записей.

Таймер обновлений маршрутов (update timer). Период этого таймера задает частоту рассылки сообщений об обновлении маршрутизации.

Тактовая частота (clock rate). Частота, с которой интерфейс последовательного канала передает биты в среду передачи данных.

Файл текущей конфигурации (`running-config file`). В коммутаторах и маршрутизаторах компании Cisco под управлением операционной системы IOS так называется файл, размещаемый в оперативной памяти устройства, в котором хранятся текущие настройки устройства.

Терминальное оборудование (Data Terminal Equipment — DTE). Устройство, расположенное на пользовательском конце интерфейса "пользователь–сеть", которое может выступать в качестве источника данных, получателя данных или и того и другого. Устройство DTE соединяется с сетью данных через устройство DCE (например, модем) и для синхронизации зачастую использует временные сигналы, генерируемые устройством DCE. Терминальное оборудование включает в себя такие устройства, как компьютеры, трансляторы протоколов и мультип-

лексоры. С точки зрения провайдера устройство DTE находится вне сети провайдера услуг и обычно представляет собой маршрутизатор.

Терминальное оборудование канала передачи данных (Data Circuit-terminating Equipment — DCE). Устройство, используемое для конвертирования данных пользователя из цифрового формата DTE в форму, приемлемую для оборудования служб распределенной сети.

Тестовый пакет (keepalive). Сообщение, отправляемое одним сетевым устройством, которое сигнализирует другому сетевому устройству о работоспособности виртуального канала между ними.

Точка демаркации, или граница (demarc). Так называют узел, в котором кабель провайдера услуг подключается к кабельной системе организации или здания.

Точка доступа (access point). Устройство в беспроводной локальной сети, позволяющее клиентам обмениваться данными друг с другом и с остальной сетью. Точка доступа подключена к проводной локальной сети и имеет радиосвязь для беспроводных соединений.

Трансляция адресов с использованием портов (Port Address Translation — PAT). Метод трансляции, который предоставляет пользователю возможность сохранять адреса в глобальном адресном пуле, позволяя транслировать порты отправителей в соединениях TCP или диалогах UDP. Разные локальные адреса затем преобразуются в глобальные, где трансляция порта обеспечивает их уникальность.

Трансляция сетевых адресов (Network Address Translation — NAT). Механизм сокращения необходимости в глобально уникальных IP-адресах. Позволяет подключаться к Интернету организации с локально уникальными адресами за счет трансляции этих адресов в глобально маршрутизируемое адресное пространство.

Универсальный локатор ресурсов (Universal Resource Locator — URL). Стандартная схема адресации для доступа к гипертекстовым документам и другим службам через браузер в сети TCP/IP. Например, адрес `http://www.certskills.com/blog` представляет собой URL, идентифицирующий протокол (HTTP), название хоста (`www.certskills.com`), а также путь к странице (`/blog`).

Уникальный локальный адрес (unique local address). Тип одноадресатного IPv6-адреса, предназначенный для замены частных IPv4-адресов.

Упорядоченная передача данных (ordered data transfer). Сетевая функция, входящая в стек TCP/IP, в которой протокол определяет, как хост отправителя должен нумеровать пересылаемые данные и как хост получателя должен переупорядочивать блоки данных, если они пришли в неправильном порядке. Эта функция также определяет, как уничтожать блоки данных, которые не могут быть доставлены в нужном порядке.

Управление потоком (flow control). Представляет собой методику, благодаря которой не допускается ситуация, когда передающий объект переполняет данными принимающий объект. При полном заполнении буферов принимающего устройства посылающему устройству отправляется сообщение о необходимости отложить передачу данных до завершения обработки данных в буферах. Примером механизма управления потоками является метод скользящего окна в TCP.

Установка соединения (connection establishment). Процесс, в ходе которого протокол с установлением соединения создает виртуальный канал. В протоколе TCP соединение создается в результате трехэтапного согласования канала с помощью специализированных сегментов.

Учет (accounting). В терминах безопасности служба, записывающая действия пользователя, в том числе и попытки доступа.

Файл `vlan.dat`. Стандартный файл, хранящий базу данных конфигураций VLAN коммутатора Cisco.

Фильтр (filter). Обычно процесс или устройство, которое определяет, передавать или не передавать трафик дальше на основе заданных критериев, таких как адрес отправителя, получателя или протокол.

Флеш-память (flash). Специализированный тип постоянной памяти, содержимое которой может быть стерто и перезаписано.

Фрейм (frame). Логически сгруппированная информация, пересылаемая в виде блока данных канального уровня по среде сети.

Фрейм Ethernet (Ethernet frame). Включает заголовок и концевик канала связи Ethernet, а также инкапсулируемые между ними данные.

Хост (host). Любое устройство, использующее IP-адрес.

Цифровая сеть с комплексным обслуживанием (Integrated Services Digital Network — ISDN). Протокол, используемый телефонными компаниями и позволяющий передавать по телефонным сетям данные, голос и другие типы трафика. Часто используется в качестве средства доступа к Интернету, а также как средство установки резервного канала между маршрутизаторами на случай отказа основного соединения WAN.

Цифровой абонентский канал (Digital Subscriber Line — DSL). Открытая сетевая технология, обеспечивающая высокую скорость передачи на ограниченные расстояния по обычному медному проводу. Используется в качестве технологии доступа к Интернету для подключения пользователя к провайдеру.

Цифровой сигнал уровня 0 (Digital Signal level 0 — DS0). Спецификация формирования фреймов при передаче цифровых сигналов по одному каналу с полосой пропускания 64 Кбит/с для передачи одного голосового вызова в импульсно-кодовой модуляции

Цифровой сигнал уровня 1 (Digital Signal level 1 — DS1). Спецификация формирования фреймов при передаче цифровых сигналов по одному каналу с полосой пропускания 1,544 Мбит/с по линии T1 (в США) или 2,108 Мбит/с по линии E1 (в Европе). Канал этого уровня включает в себя 24 подканала DS0 по 64 и 8 Кбит/с управляющей информации для соединения T1.

Цифровой сигнальный уровень 3 (Digital signal level 3 — DS3). Канал на 44,736 Мбит/с от телефонной компании с 28 каналами DS1, а также одним служебным. Называется также T3.

Частные адреса (private addresses). Зарезервированные IP-адреса в классах сетей A, B и C, которые предназначены только для использования в локальной сети какой-либо организации. Эти адреса описаны в документе RFC 1918; они не маршрутизируются в Интернете.

Часть подсети (subnet part). В подсетях адресного пространства IPv4 — одна из трех частей адреса или маски подсети, уникальным образом идентифицирующая подсети в рамках классовой сети.

Часть хоста (host part). Термин, описывающий часть адреса формата IPv4, которая уникальным образом идентифицирует хост в подсети. Часть хоста в адресе идентифицируется двоичными нулями в маске подсети.

Четырехпроводной канал (four-wire circuit). Линия от телекоммуникационной компании, состоящая из двух витых пар. Каждая из пар используется для передачи сигнала в одном направлении, за счет чего достигается дуплексный режим работы.

Шаблон маски (wildcard mask). Маска, используемая командами ACL Cisco IOS и OSPF и командой **network** протокола EIGRP.

Шина (bus). Набор электрических цепей, через которые передаются данные от одной части компьютера другой.

Ширина полосы пропускания (bandwidth). Объем информации, проходящей через сетевое соединение за определенный период времени. Сам термин пришел из ранних стандартов телекоммуникационных технологий, когда таким термином описывался частотный диапазон или ширина частотного диапазона устройств для передачи данных.

Широковещательная подсеть (broadcast subnet). Разделяя сеть класса A, B или C на подсети, инженер должен выделить адрес, в части хоста которого все биты равны 1. Такой адрес и является широковещательным, а для последней подсети он совпадает с широковещательным адресом классовой сети. Последнюю подсеть классовой сети зачастую называют широковещательной.

Широковещательный адрес (broadcast address). В общем, любой адрес, представляющий все устройства. Применяется для передачи одного сообщения на все устройства. В Ethernet — это MAC-адрес со всеми двоичными единицами или FFFF.FFFF.FFFF в шестнадцатеричном виде. Для протокола IPv4 применяется *широковещательный адрес подсети* (subnet broadcast address).

Широковещательный адрес локальный подсети (local subnet broadcast address). IPv4-адрес 255.255.255.255. Пакет, посланный на этот адрес, передается как широковещательный пакет канала связи, но только на хосты той подсети, в которую он был первоначально послан. Маршрутизаторы не перенаправляют эти пакеты.

Широковещательный адрес подсети (subnet broadcast address). Специализированный адрес в каждой из подсетей, представляющий собой наибольший адрес блока для подсети. Адрес работает таким образом, что если пакет отправлен на широковещательный адрес, то его получат все устройства в подсети.

Широковещательный адрес сети (network broadcast address). В стандарте IPv4 — специальный адрес в каждой классовой или бесклассовой сети, используемый для широковещательной пересылки пакета всем хостам в сети. В числовом выражении такой адрес представляет собой наибольший адрес в сети, например, для классовой частной сети 10.0.0.0 широковещательный адрес будет равен 10.255.255.255.

Широковещательный домен (broadcast domain). Совокупность всех устройств, которые получают широковещательные фреймы от любого из устройств этой совокупности. Границы широковещательного домена обычно определяются маршрутизаторами (в коммутируемых сетях — виртуальными сетями VLAN), поскольку маршрутизаторы не пересылают широковещательные фреймы.

Широковещательный фрейм (broadcast frame). Фрейм, рассылаемый всем хостам сегмента локальной сети, адрес получателя которого равен FFFF.FFFF.FFFF.

Шифрование (encryption). Применение специального алгоритма для изменения внешнего вида данных таким образом, чтобы их содержание было непонятно для тех, кому не предоставлены соответствующие средства дешифрования.

Экранированная витая пара (Shielded Twisted Pair — STP). Тип кабеля витой пары, в которой каждая пара имеет свой экран, а весь кабель защищен общим экраном.

Экспресс-передача Cisco (Cisco Express Forwarding — CEF). Метод внутренней обработки маршрутизаторов Cisco, существенно повышающий эффективность процесса маршрутизации за счет кеширования маршрутов IP в таблице, поиск в которой осуществляется очень быстро. Заголовок канала связи тоже кешируется, а не создается заново для каждого передаваемого пакета.

Энергонезависимая память (NonVolatile RAM — NVRAM). Оперативное запоминающее устройство, содержимое которого сохраняется при отключении питания.

Эталонная модель взаимодействия открытых систем (Open System Interconnection — OSI reference model). Структурная модель сети, разработанная Международной организацией по стандартизации (ISO). Эта модель включает в себя семь уровней, каждый из которых выполняет свои специфические функции, такие как адресация, управление потоком, контроль ошибок, инкапсуляция и надежная передача сообщений. Эталонная модель OSI используется как универсальный метод обучения сетевых специалистов для понимания ими функций компьютерной сети.

Язык гипертекстовой разметки (HyperText Markup Language — HTML). Простой язык гипертекстового форматирования документов, в котором используются теги (неотображаемый текст разметки документа) для указания способа представления частей документа приложениями просмотра, такими, как веб-браузеры.

Предметный указатель