**К. Е. Самуйлов, И. А. Шалимов,
Н. Н. Васин, В. В. Василевский,
Д. С. Кулябов, А. В. Королькова**

Сети и системы передачи информации: телекоммуникационные сети

Учебник

Рекомендовано Государственным образовательным учреждением высшего профессионального образования «Академия Федеральной службы безопасности Российской федерации» в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальностям направления подготовки 090302 «Информационная безопасность телекоммуникационных систем».
Регистрационный номер рецензии № 742 от 25 февраля 2010 г. (ГОУ ВПО «Московский государственный университет печати»).

Москва
Издательство Юрайт
2015

УДК 004.72:004.057.4(075.8)
ББК
С

Рецензенты:
доктор технических наук, профессор, декан факультета «Радио и телевидение» МТУСИ Пестряков Александр Валентинович,
доктор технических наук, профессор кафедры «Сетей связи и передачи данных» СПбГУТ им. проф. М.А. Бонч-Бруевича Кучерявый Андрей Евгеньевич
кандидат технических наук, доцент, заведующий кафедрой «Информационная безопасность систем и технологий» ПГУ Зефиров Сергей Львович

Самуйлов К. Е.

С Сети и системы передачи информации: телекоммуникационные сети : Учебник / К. Е. Самуйлов, И. А. Шалимов, Н. Н. Васин, В. В. Василевский, Д. С. Кулябов, А. В. Королькова. — Москва : Издательство Юрайт, 2015. — 422 с. : ил.

В учебнике последовательно изложены основные концепции современного состояния сетей и систем передачи информации. Рассматриваются аспекты и уровни организации сетей – от физического до уровня приложений модели взаимодействия открытых систем. Теоретический материал дополнен лабораторным практикумом и практическими заданиями.

Данный учебник рекомендуется Учебно-методическим объединением УМО по образованию в области информационной безопасности к использованию в образовательных учреждениях, реализующих образовательные программы высшего профессионального образования, по дисциплине «Сети и системы передачи информации» по специальности 090302 «Информационная безопасность телекоммуникационных систем».

УДК 004.72:004.057.4(075.8)
ББК

ISBN

Предисловие

Представляемый внимаю читателей учебник направлен на достижение следующих целей:

- Ввести учащихся в предметную область существующих систем и сетей телекоммуникаций.
- Сформировать понятийный аппарат в области концепций, архитектур, стандартов современных систем и сетей телекоммуникаций.
- Ознакомить слушателей с современными технологиями в области систем и сетей телекоммуникаций.
- Создать у слушателей понимание принципов построения современных систем и сетей телекоммуникаций.

Учебник является обзорным. Читатель должен получить целостное представление о существующих сетевых технологиях, увидеть их генезис. Авторы являются скорее практиками, чем теоретиками.

Основной упор делается на описании протоколов. Для понимания путей и движущих сил развития протоколов, авторы предлагают ознакомиться не только с современным состоянием дел в этой области, но и дают исторический обзор.

Структурирование книги идёт по нескольким направлениям. Во-первых, можно выделить две части (интерферирующие друг с другом): протоколы компьютерных сетей и мультисервисные сети.

Первая часть изучает протоколы компьютерных сетей согласно структуре эталонной модели взаимодействия открытых систем. Тут уделяется основное внимание двум стекам протоколов — Ethernet и TCP/IP. Другие стеки протоколов преподносятся как имеющие скорее теоретическое значение, нежели практическое. Вторая часть рассматривает современное состояние сетей телекоммуникаций.

В первой главе даются и объясняются такие базовые понятия систем телекоммуникаций, как протокол, интерфейс, служба. Даётся обзор существующих сетей связи, сетевых сервисов. Рассматривается структура и основные аспекты деятельности стандартизирующих организаций.

Во второй главе рассматриваются общие принципы построения модели взаимодействия открытых систем (OSI), иерархия протоколов различных стеков протоколов (TCP/IP, IEEE, ISO/OSI, H.323, SS7 и др.) по отношению к модели OSI.

В третьей главе рассматриваются методы и технологии физического уровня модели OSI, расширенного в духе IEEE 802. В частности, даётся обзор возможных сред передачи (в том числе и стандарты кабельной системы), методов кодирования сигнала и сферы их применения.

В четвёртой главе изучаются методы и протоколы доступа к среде (семейства ALOHA и CSMA), а также технологии сетей (Ethernet,

Fast Ethernet, Gigabit Ethernet, Token Ring, 100VG–AnyLAN, Wireless Networks, WiMAX, Bluetooth, FDDI, ISDN, Frame Relay). Упор делается на стандарты IEEE 802.x, особенно на IEEE 802.3 (Ethernet).

В пятой главе рассматриваются протоколы межсетевого уровня стека протоколов TCP/IP. Особое внимание уделяется протоколу IP: изучается формат кадра IP, фрагментация IP, IP-адресация (IPv4 и IPv6), взаимодействие межсетевого уровня с физическим. В этой же главе отдельным пунктом изучается проблема маршрутизации: классификация алгоритмов маршрутизации и, собственно, протоколы статической (iproute2, click) и динамической маршрутизации (RIP, OSPF, BGP, IGRP), сфера их применения, достоинства и недостатки. Кратко рассматриваются другие протоколы межсетевого уровня стека протоколов TCP/IP (ARP, RARP, ICMP, IGMP), их назначение.

В шестой главе рассматриваются протоколы транспортного уровня стека протоколов TCP/IP: TCP (формат TCP-пакета, алгоритм установления связи (сессия TCP), надежная доставка, технология скользящего окна, процедура «медленный старт»), UDP (формат UDP-пакета, псевдозаголовок, ненадежная передача), DCCP, SCTP.

В седьмой главе рассматриваются вопросы обеспечения информационной безопасности сетей передачи данных. Дается краткий обзор существующих угроз, рассматриваются элементы обеспечения защиты сетевого оборудования.

Восьмая глава посвящена мультисервисным сетям. Также в ней описываются основные подходы к построению сетей следующего поколения (NGN). В исторической ретроспективе рассматриваются два основных подхода к построению конвергентных сетей, называемых также сетями следующего поколения (Next Generation Networks, NGN), — Softswitch и IMS. Даётся сравнительный обзор концепций Softswitch и IMS как развитие концепций конвергирования сетей коммутации каналов (в частности, ТфОП) и сетей коммутации пакетов (в частности, IP-сети). Описываются архитектурные особенности и основные протоколы обоих подходов.

В результате освоения материалов данного учебника студент должен:

знать:

- основные термины и определения предметной области сетей связи;
- методы коммутации информации в сетях связи;
- основные стандартизирующие организации в области сетей связи и сферу их деятельности;
- эталонную модель взаимодействия открытых систем, модели TCP/IP, IEEE802;
- перечень основных протоколов различных стеков сетей и систем передачи информации;

– основные типы сред передачи данных, их характеристики, область их применения;
– типы модуляции аналоговых сигналов, основные методы кодирования сигналов;
– основные модели, технологии и протоколы доступа к среде передачи данных, структуру протоколов доступа к среде;
– структуры заголовков протоколов межсетевого уровня стека TCP/IP;
– принципы организации адресного пространства IPv4 и IPv6;
– принципы коммутации по меткам (технология MPLS);
– принципы построения сетей передачи данных и настройки сетевого оборудования;
– структуры заголовков протоколов транспортного уровня стека TCP/IP, принципы передачи данных на транспортном уровне;

уметь:

– грамотно пользоваться языком предметной области сетей связи;
– зная основные функции протоколов различных стеков, определять их место в модели взаимодействия открытых систем;
– определять элементы структурированной кабельной системы организации, ограничения их применения;
– строить временные диаграммы информационных сигналов;
– ориентироваться в стандартах IEEE 802;
– оценивать работоспособность сети, построенной на базе технологии Fast Ethernet;
– планировать адресное пространство IPv4, определять тип адреса IPv6;
– определять область применения того или иного транспортного протокола;

владеть:

– базовыми знаниями в области современных сетей и систем передачи информации;
– навыками анализа основных характеристик уровней модели взаимодействия открытых систем;
– навыками анализа основных характеристик протоколов различных стеков;
– способностью применять математический аппарат, в том числе с использованием вычислительной техники, для решения профессиональных задач, например, для оценки работоспособности сети, построенной на базе технологии Fast Ethernet;
– способностью планировать структуру сети передачи данных;
– способностью настраивать коммутационное и маршрутизирующее оборудование;
– способностью анализировать данные заголовков сетевых протоколов и протоколов транспортного уровня.

Глава 1. Общие сведения о сетях и системах передачи информации

Данная глава посвящена введению в предметную область сетей связи. В ней рассматриваются основные понятия, связанные с сетевыми технологиями. Основу главы составили материалы авторов из источников [1—3].

В результате освоения данной темы студент должен:

знать:

- основные термины и определения предметной области сетей связи;
- методы коммутации информации в сетях связи;
- основные стандартизирующие организации в области сетей связи и сферу их деятельности;

уметь:

- грамотно пользоваться языком предметной области сетей связи;

владеть:

- базовыми знаниями в области современных сетей и систем передачи информации.

1.1. Основные термины и определения

Сеть связи можно рассматривать как систему, состоящую из линий связи, соединяющих некоторое множество узлов, и предназначенную для передачи данных от одного узла к другому с определёнными параметрами качества обслуживания.

Основу сетей связи составляют *многоканальные системы передачи* (*МСП*), для передачи информации используются электромагнитные сигналы. Структурная схема МСП приведена на рис. 1.1.

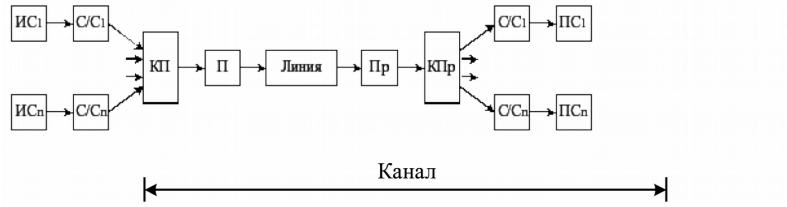


Рис. 1.1. Структурная схема МСП

На рис. 1.1 обозначено: $ИС_i$ — источники передаваемых сообщений; $С/С_i$ — преобразователи сообщения в первичный сигнал на передающей стороне и полученного первичного сигнала в сообщение на

приёмной стороне; КП — коммутатор передающей стороны; КПр — коммутатор приёмной стороны; П — преобразователь первичного сигнала в линейный сигнал (передатчик), Пр — преобразователь линейного сигнала в первичный (приёмник); $ПС_i$ — получатели сообщений.

Сообщения абонентов при помощи $С/С_i$ преобразуются в *первичные электрические сигналы*. Множество сформированных сигналов коммутируются на передающей стороне коммутаторами КП и на приёмной стороне коммутаторами КПр, чтобы установить соединение источника сообщения $ИС_j$ с соответствующим получателем $ПС_j$.

Первичные электрические сигналы при помощи преобразователя П преобразуются во вторичные или *линейные сигналы*, характеристики которых хорошо согласуются с параметрами линии связи. На приёмной стороне Пр принимает и преобразует линейные сигналы в первичные.

Наиболее дорогостоящим элементом МСП является *линия связи*, поскольку её протяжённость может составлять сотни и даже тысячи километров.

Под линией связи (каналом связи) обычно понимают некую физическую среду распространения сигналов, связывающую некоторое множество технических устройств, обеспечивая при этом передачу с сигналов на заданное расстояние определёнными показателями: полосой частот, скоростью передачи и т.п. Например, линию связи может образовывать кабельная система на базе оптоволокна или меди, или радиоканал (беспроводная среда передачи).

Линии связи (*каналы связи*) необходимы для того, чтобы множество одновременно передаваемых сигналов не мешали друг другу в общей линии многоканальной системы. На рис. 1.1 каналы образуются оборудованием коммутаторов и формируются в среде общей линии связи.

Каналы связи разделяют на *непрерывные (аналоговые)* и *дискретные (цифровые)*. Кроме того, каналы связи могут иметь один из трёх типов направленности передачи информации [1]:

- *симплексный (Simplex Transmission)* — передача данных в одном направлении;
- *полудуплексный (Half-Duplex Transmission)* — передача данных в двух направлениях, но только в одном направлении в каждый момент времени;
- *дуплексный (Duplex Transmission)* — передача данных одновременно в двух направлениях.

Сеть передачи информации образуется путём соединения между собой множества источников и приёмников сообщений (абонентов). Соединение обеспечивается аппаратными средствами многоканальных систем и средой передачи сигналов (линиями связи).

Аппаратуру абонентов принято называть узлами или *конечными устройствами* (У) сетей, которым соответствует широко распространенное англоязычное наименование *Host* (хост). Соединение многочис-

ленных узлов (абонентов), находящихся на большом расстоянии друг от друга, обычно обеспечивается через транзитные (промежуточные) *сетевые элементы* (СЭ) (рис. 1.2).

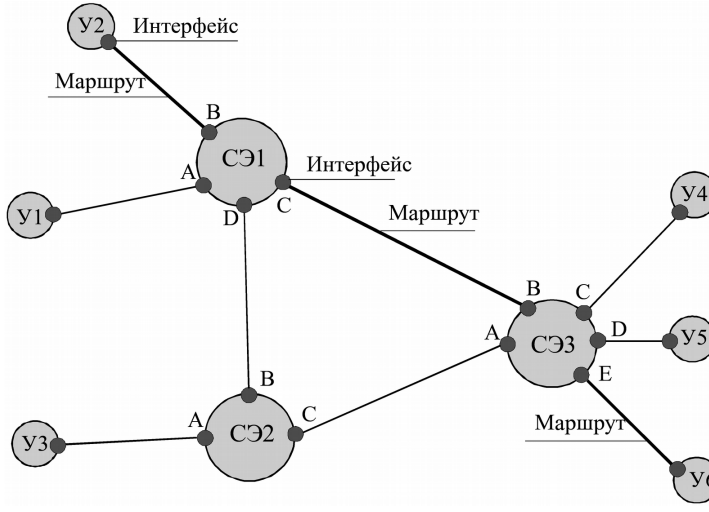


Рис. 1.2. Сеть передачи информации [3]

Таким образом, *сеть передачи информации* представляет собой совокупность узлов (У) и сетевых элементов (СЭ), соединённых линиями (каналами) связи. При этом сетевые элементы производят перемещение (*коммутацию*) поступившего сообщения с одного интерфейса (входного порта) на другой (выходной порт). Например, в сети рис. 1.2 при передаче сообщения от узла У2 узлу У6 сетевой элемент СЭ1 производит коммутацию сообщения с входного интерфейса B на выходной интерфейс C, сетевой элемент СЭ3 — с входного интерфейса B на выходной E. Таким образом, формируется *маршрут*, по которому осуществляется передача сообщений.

В некоторых сетях все возможные маршруты заранее созданы и необходимо только выбрать оптимальный. Процесс выбора оптимального маршрута получил название *маршрутизация*, а устройство её реализующее — *маршрутизатор*. Таким образом, промежуточные сетевые элементы могут выполнять функции коммутаторов, которые формируют маршрут, и (или) маршрутизаторов, которые производят выбор оптимального маршрута.

Для выбора (создания) маршрута необходимо задать *адреса* источника и получателя сообщения. Выбор оптимального маршрута

сетевые элементы (маршрутизаторы) производят на основе *таблиц коммутации* (или *маршрутизации*) с использованием определенного критерия — *метрики*.

При анализе систем передачи информации рассматривается процесс передачи сигналов между сетевыми элементами и формирование каналов для передачи многих информационных потоков, а при анализе сетей передачи информации — процессы коммутации и маршрутизации, т.е. передачи *информационных потоков* (*трафика сообщений*) по каналам (*трактам*) линий сети связи.

Информационный поток — это совокупность передаваемых сообщений, или последовательность информационных единиц, объединённых общими признаками.

1.2. Понятие протокола. Иерархия протоколов. Интерфейсы и сервисы

Под сетевым протоколом обычно понимают совокупность правил взаимодействия двух элементов сети при обмене информацией между ними.

Большинство протоколов имеют иерархический принцип организации (рис. 1.3): нижележащий уровень предоставляет через *интерфейс* некоторый набор услуг (сервисов) вышележащему уровню, не раскрывая детали реализации предоставляемой услуги.

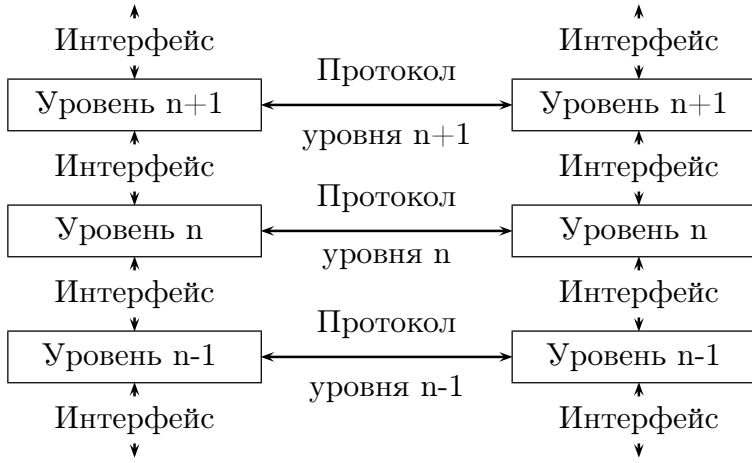


Рис. 1.3. Уровни протоколов [1; 2]

Правила и соглашения, используемые при взаимодействии уровня n одного узла и уровня n другого узла называются *протоколом уровня n*.

На каждом уровне имеется активный элемент *(сущность (Entity))*. *Одноранговыми сущностями* называют сущности одного уровня на разных узлах сети. Сущности уровня n предоставляют услуги уровню $n+1$, т.е являются *поставщиками услуг*. При этом сущности уровня $n+1$ являются *потребителями услуг* уровня n. Оказывая услуги уровню $n+1$, уровень n может использовать услуги (быть потребителем) нижележащего уровня $n-1$.

Сервис (услуга) определяет операции, которые будет выполнять уровень, но не реализацию этих операций. Сервис по сути описывает интерфейс взаимодействия двух смежных уровней — поставщика услуг (нижележащего уровня) и потребителя (вышележащего уровня).

Доступ к услугам некоторого уровня обеспечивается через *точки доступа к услуге (Service Access Point, SAP)*, т.е. посредством набора правил используемого для взаимодействия между уровнями интерфейса. Через SAP сущность вышележащего уровня передаёт сущности нижележащего уровня *элемент данных интерфейса (Interface Data Unit, IDU)*, который состоит из *элемента данных услуги (Service Data Unit, SDU)* и некоторой *управляющей информации (Interface Control Information, ICI)*. При этом SDU может быть разбит на несколько фрагментов. Тогда его пересылка осуществляется в виде отдельных *элементов данных протокола (Protocol Data Unit, PDU)* или *пакетов*.

Таким образом, *протокол* определяет набор правил, которые описывают формат и назначение пакетов, передаваемых между одноранговыми сущностями внутри уровня. По сути протокол определяет услуги уровня, на котором он работает. Протокол может претерпеть изменения, но предоставляемые услуги не должны меняться.

Протоколы разделяют на протоколы с *установлением соединения (Connection Oriented)* и без установления соединения (Connectionless). В первом случае до начала обмена данными отправитель и получатель должны установить соединение, определив предварительно некоторые параметры протокола. После завершения передачи данных (завершения сеанса передачи) соединение должно быть разорвано с помощью обмена специальными управляющими сообщениями.

Во втором случае передача данных осуществляется без организации специальной процедуры по установлению соединения с получателем отправляемых данных.

Список протоколов, который использует элемент сети, называется *стеком протоколов*. Совокупность уровней и протоколов образуют *архитектуру сети*.

1.3. Обобщённая структурная схема сети

Архитектура сети связи является одной из основных характеристик, определяющих состав сети, раскрывающий типы образующих её функциональных компонентов, иерархию и характер их взаимодействия.

В связи с большим разнообразием видов передаваемых сообщений и сигналов, среды распространения, методов и устройств коммутации или маршрутизации сигналов и информационных потоков архитектура сетей связи классифицируется согласно требованиям *Единой сети электросвязи Российской Федерации* (*ЕСЭ РФ*).

ЕСЭ РФ определяется совокупностью сетей связи различного назначения и технологий, располагающихся на территории РФ.

Модель архитектуры сети связи, предложенная в положении о единой сети электросвязи (ЕСЭ) РФ, может быть представлена в виде, показанном на рис. 1.4.

Первый уровень модели — *первичная сеть* (первичные сети), образуемая на системах передачи определённых родов связи. Первичные сети разделяются на *магистральные*, *внутризоновые* и *местные* (городские и сельские). Первичная сеть представляет собой совокупность всех каналов связи не зависимо от назначения и вида связи; она включает линии связи и каналообразующую аппаратуру.

Второй уровень – *вторичные сети*, образуемые на базе каналов передачи первичной сети и систем коммутации, выполняющих функции распределения сообщений по заданному адресу. Вторичные сети различаются по виду передаваемых по ним сообщений: *телефонные*, *передачи данных*, *телеграфные*, *передачи газет*, *звукового вещания*, *телевизионного вещания* и др. При интеграции сетей связи вторичные сети превращаются в единую сеть, обеспечивающую передачу и распределение сообщений различных видов связи (*передачи речи*, *данных*, *факсимильных сообщений* и др.).

Третий уровень модели – *службы связи*, обеспечивающие предоставление пользователям услуг различных видов связи.

Четвертый уровень — *пользователь услуг связи*. Он определяется видом связи (передача речи, телеграфных и/или факсимильных сообщений, сообщений данных), а также терминальным оборудованием, имеющимся у пользователя.

При классификации сетей по категориям сети ЕСЭ подразделяются на *сети общего пользования*, *выделенные сети*, *технологические сети* и *сети связи специального назначения*.

В соответствии с выполняемыми функциями сети ЕСЭ разделяются на *сети доступа* и *транспортные сети*. По транспортной сети передаются высокоскоростные (широкополосные) потоки информации. Транспортная сеть связи включает магистральную (междугородную и международную) и зоновые (региональные) сети связи. Сеть доступа

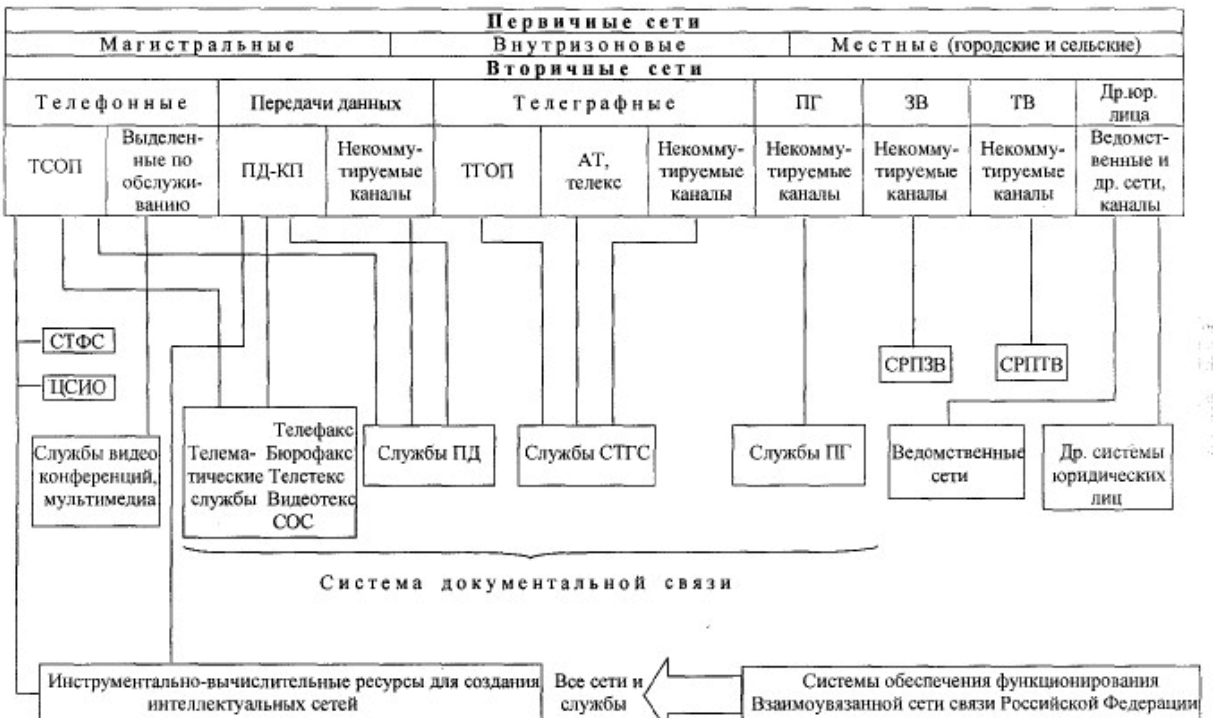


Рис. 1.4. Модель архитектуры ЕСЭ России

обеспечивает доступ абонентов к транспортной сети; она также называется *сетью абонентского доступа* и по территориальному признаку является местной сетью. Данная сеть состоит из абонентских линий и оконечных устройств.

Обобщённая структурная схема телекоммуникационной сети включает в себя *транспортный уровень* (магистральная сеть), *уровень доступа* (сети доступа) и *терминальное оборудование пользователей*.

Компоненты телекоммуникационной сети:

- магистральной сети;
- сетей доступа;
- терминального оборудования пользователей;
- информационных центров, или *центров управления сервисами* (*Services Control Point, SCP*).

Магистральная сеть объединяет отдельные сети доступа, обеспечивая транспорт трафика между ними по высокоскоростным каналам. По сути магистральные сети относятся к глобальным сетям связи (Wide Area Network, WAN).

Сеть доступа располагается на нижнем уровне иерархии телекоммуникационной сети и предназначена для агрегации потоков, поступающих по различным каналам связи от клиентского оборудования, в магистральной сети.

Сеть доступа по сути представляет собой региональную сеть большой разветвлённости. Она может быть многоуровневой. Сетевые элементы нижнего уровня мультиплексируют информацию, поступающую по многочисленным абонентским каналам (абонентскими окончаниями), и передают её сетевым элементам верхнего уровня для перенаправления элементам магистрали. Размер сети доступа определяет число её уровней — небольшая сеть доступа будет иметь один уровень, крупная — несколько.

В компьютерной сети *оконечным оборудованием* являются компьютеры, в телефонной — телефонные аппараты, в телевизионной или радиосети — соответствующие теле- или радиоприёмники.

Оконечное оборудование пользователей может формировать сеть, не входящую в состав телекоммуникационной сети. Например, совокупность компьютеров пользователей организации образует *локальную сеть (Local Area Network, LAN)*. Локальные сети характеризуются высокой скоростью передачи данных на сравнительно небольшие расстояния.

Информационные центры (центры управления сервисами) предоставляют информационные сетевые услуги. В таких центрах хранится пользовательская информация (информация, непосредственно интересующая конечных пользователей) и служебная информация, помогающая поставщику услуг предоставлять услуги пользователям.

Пользовательская информация обычно содержит разнообразную справочную и новостную информацию. Подобные центры телефонных

сетей оказывают, например, услуги экстренного вызова милиции или скорой помощи, а также справочные услуги различных организаций и предприятий — вокзалов, аэропортов, магазинов и т. п.

К служебной информации обычно относят различные данные системы авторизации и аутентификации пользователей, с помощью которых организация, которая владеет сетью, проверяет права пользователей на получение тех или иных услуг. Это могут быть системы биллинга, используемые для определения платы за предоставляемые услуги, или базы данных, содержащие учётные записи пользователей и перечни предоставляемых пользователям услуг.

Сети конкретного типа обладают своими особенностями, в них могут отсутствовать некоторые элементы обобщённой сети, но в целом их структура соответствует описанной выше.

1.4. Методы коммутации информации в сетях связи

Основными задачами, решаемыми системами коммутации в сетях передачи информации являются:
– распределение информации;
– предоставление абонентам различного типа услуг связи;
– объединение элементов сети связи в единую систему.

Оборудование систем коммутации вместе с программным обеспечением образуют *коммутационный центр (КЦ)* сети связи.

Распределение информации может осуществляться или путём распределения каналов, или распределением информационных потоков по направлениям связи. В сетях связи информация распределяется в процессе *коммутации*.

Коммутацией называется совокупность операций, при выполнении которых сигнал, поступивший на вход элемента сети связи, называемого *системой коммутации*, поступает на её выход с изменённым идентификатором пункта назначения. Иначе под коммутацией в какой-либо системе понимается изменение координаты информационного сигнала на выходе этой системы по отношению к его координате на её входе. Координаты могут быть пространственные, временные, спектральные, фазовые и т. п. В настоящее время применяются пространственная и временная коммутация.

Различают два вида коммутации: *долговременную* и *оперативную*. Под долговременной понимается коммутация, выполняемая вне зависимости от поступления заявки от источника информации на передачу одиночного сообщения. Коммутация этого вида осуществляется по плану (схеме) связи или по команде от системы (лица) управления связью.

Оперативной называется коммутация, осуществляемая по заявке источника информации (абонента) на установление соединения для

передачи одного сообщения. После передачи каждого сообщения (по сигналу «отбой») установленное соединение разрушается. Различают несколько типов оперативной коммутации. В сетевых технологиях наиболее часто находят применение *коммутация каналов* и *коммутация пакетов* [1; 2]:

- *коммутация каналов (Circuit Switching)* — режим передачи, при котором на время передачи информации (до разъединения соединения) формируется составной канал (соединение), проходящий через несколько транзитных узлов;
- *коммутация сообщений (Messege Switching)* — режим передачи, при котором осуществляется приём, хранение, выбор исходящего направления и дальнейшая передача сообщений без нарушения их целостности;
- *коммутация пакетов (Packet Switching)* — режим передачи сообщений, при котором сообщения разбиваются на пакеты ограниченного размера, причём канал передачи занят только во время передачи пакета и освобождается после её завершения;
- *коммутация ячеек (Cell Switching)* — режим передачи пакетов фиксированного размера.

В сети связи между двумя коммутационными центрами (КЦ) могут использоваться простые либо составные каналы связи (каналы передачи). Простой канал связи (передачи) образуется на базе одной системы передачи и состоит из двух комплектов каналообразующей аппаратуры (КОА), соединённых направляющей системой (НС) или, по-другому, средой распространения сигналов электросвязи. Составной канал связи представляет собой два и более последовательно соединённых простых каналов (рис. 1.5).

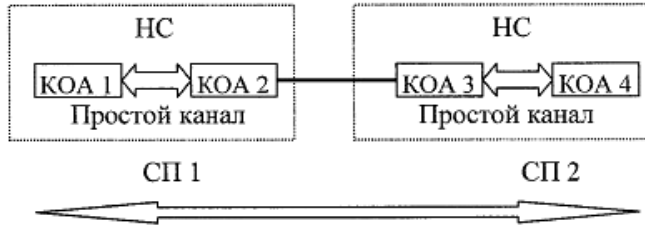


Рис. 1.5. Структура составного канала

Коммутация каналов может осуществляться по заявке абонента (источника информации) на время передачи одного сообщения (оперативная коммутация каналов) либо оператором связи по плану или по запросу другого оператора вне зависимости от конкретной заявки на передачу сообщений (долговременная или кроссовая коммутация

каналов). В первом случае примерами могут служить соединения между абонентами на местной либо междугородной телефонных сетях; во втором случае — закрепление оконечной аппаратуры связи за конкретным типовым каналом связи или образование составного канала на кроссе узла связи.

Так как созданный (скоммутированный) канал связи выделяется в полное распоряжение пары абонентов, то он обеспечивает требуемое качество передачи сообщения. Например, технологии коммутации каналов широко используются в телефонных сетях общего пользования (ТфОП). Однако при этом каналы связи используются не эффективно. Паузы между словами и, особенно, между фразами могут быть достаточно большими. Поэтому коэффициент использования канала часто оценивают величиной 0,25.

В отличие от сетей с коммутацией каналов сети с коммутацией пакетов могут более эффективно использовать свои ресурсы. При передаче информации важно обеспечить *надёжность* её доставки получателю. Надёжность обеспечивается не только за счёт использования надёжных аппаратных средств связи, но и за счёт повторной передачи сообщения в случае его потери или искажения. Поэтому большие сообщения делятся на сравнительно небольшие *пакеты*. При потере части сообщения повторно передается не всё сообщение, а только потерянный или искажённый пакет.

Каждому пакету присваивается адрес получателя информации и признак принадлежности пакета к данному сообщению. В ряде случаев пакету присваивается его порядковый номер в данном сообщении. Формат пакета включает две части: *служебную* и *информационную* (рис. 1.6). Служебная часть пакета и содержит адрес получателя сообщения, а также другую служебную информацию.

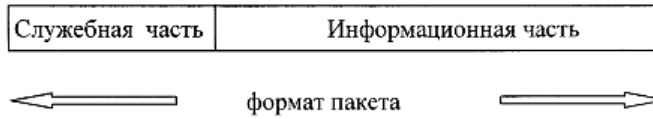


Рис. 1.6. Формат пакета при коммутации пакетов

По внешнему виду каждый пакет формально представляет собой независимое сообщение.

Для исключения потерь передаваемых по сети пакетов данных сетевые устройства должны иметь *буферы памяти* на своих интерфейсах, с тем чтобы в них хранились пакеты, ожидающие своей очереди на обработку. Для обеспечения требуемой надёжности передачи сообщений по сети наряду с повторной передачей пакетов данных и использования буферов памяти на интерфейсах, сеть должна работать

в недогруженном режиме, когда средняя интенсивность поступления данных будет меньше средней интенсивности их обработки.

Примером сетей с коммутацией пакетов являются компьютерные сети, которые создавались для передачи цифровых данных.

В создаваемых в настоящее время мультисервисных *сетях следующего поколения (Next Generation Network, NGN)* технология коммутации пакетов используется для передачи всех видов трафика (аудиосигналов IP-телефонии, видео информации, компьютерных данных). Однако требования к *качеству обслуживания (Quality of Service, QoS)* разных видов передаваемого трафика будут различны.

Например, при передаче по сети с коммутацией пакетов трафика аудио-сигналов IP-телефонии, который чувствителен к задержкам пакетов и вариации задержек, указанные параметры должны быть минимизированы. Каналы передачи видео-информации должны характеризоваться малыми задержками и возможностью передачи большого объёма данных с высокой скоростью. При передаче компьютерных данных, которые слабо чувствительны к задержкам, однако очень чувствительны к потерям и искажениям пакетов (сообщений), важно обеспечить *надёжность*, за счёт повторной передачи потерянных или искажённых пакетов.

Характерным для коммутации пакетов является «эстафетная» передача пакета от терминала связи (ТС) источника информации к коммутационному центру КЦ1, далее от КЦ1 к КЦ2, от КЦ2 к КЦ3 и т. д. к терминалу связи потребителя информации (рис. 1.7).

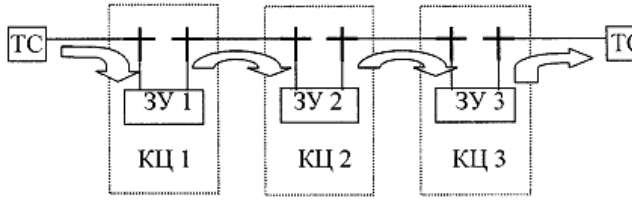


Рис. 1.7. Принцип коммутации пакетов

В зависимости от принятых на сети связи алгоритмов передачи и коммутации пакетов различают *виртуальные* или *дейтаграммные* соединения или сети. При реализации виртуальных соединений все пакеты одного сообщения передаются по одному пути, определённому при установлении соединений для первого пакета данного сообщения. В этом случае для сокращения времени передачи сообщения во всех пакетах, кроме первого, информация о номере вызываемого абонента из служебной части пакета может быть исключена (остается лишь

признак принадлежности пакета к данному сообщению). Примером сетей с виртуальными каналами являются сети Frame Relay, ATM.

При дейтаграммном способе передачи пакетизированных сообщений каждый пакет ищет на сети связи свой (оптимальный на момент его поступления) путь от источника к потребителю информации. Естественно, что при этом каждый из них должен в своей служебной части содержать полный адрес вызываемого абонента. На оконечном (приёмном) пункте все пакеты объединяются (агрегируются, сшиваются) в одно сообщение. Однако при этом следует учесть, что ввиду независимости прохождения пакетов по сети связи со сложной структурой, может случиться, что ранее поступивший пакет может выбрать более длинный путь, чем пакет, поступивший после него, который выбрал более короткий путь. В таком случае может возникнуть ситуация, при которой на конечный пункт сети раньше придёт пакет, переданный позже. То есть пакеты будут приняты не в том порядке, в котором передавались. Для правильной «сшивки» сообщения из поступающих пакетов каждый из них в служебной части должен содержать свой идентификационный номер в данном сообщении. На «сшивку» сообщения отводится определённое время, в течение которого все поступившие из сети пакеты распределяются по их порядковым номерам. Пакет, поступивший по истечении этого времени, теряется. Примером дейтаграммных сетей являются IP-сети.

Использование того или иного вида или типа коммутации принципиально не зависит ни от вида связи (телефонная, телеграфная, факсимильная, передачи данных и др.), ни от формы образующих сообщения сигналов (аналоговые, дискретные). Однако при выборе вида и типа коммутации необходимо учитывать требования к качеству обслуживания поступающих сообщений.

Эффективность сетей с пакетной коммутацией выше, чем сетей с коммутацией каналов, что предопределило использование сетей с коммутацией пакетов для передачи всех видов трафика, т.е. мультисервисных сетей следующего поколения (NGN).

В сетях следующего поколения обеспечивается конвергенция существующих на данный момент сетей в единую информационную сеть связи, базирующуюся на сетях передачи данных. Пользователям сети NGN предоставляется широкий спектр услуг с гарантированным качеством.

1.5. Основные технологии сетей передачи данных

Сети телекоммуникаций можно классифицировать[1] по нескольким параметрам [1; 2]:

[1]Заметим, что мы строим классификацию, а не таксономию.

1. по размеру сети:
   - *локальные сети (Local Area Network, LAN)* — сети здания или организации;
   - *региональные сети (Metropolitan Area Network, MAN)* — сети уровня города или региона;
   - *глобальные сети (Wide Area Network, WAN)* — сети, охватывающие большие территории и включающие в себя десятки и сотни тысяч компьютеров;
2. по типу коммутации:
   - сети с коммутацией пакетов (TCP/IP, IPX/SPX, ATM, сети сотовой связи 3G);
   - сети с коммутацией каналов (ТфОП, DECT[1], сети сотовой связи 1G и 2G);
   - смешанные (сети сотовой связи 2,5G);
3. по установлению виртуального канала:
   - с установлением виртуального канала (сети X.25, Frame Relay, ATM, ТфОП, DECT);
   - без установления виртуального канала (сети TCP/IP, IPX/SPX);
4. по используемому стеку протоколов;
5. по количеству используемых стеков протоколов:
   - монопротокольные сети;
   - мультипротокольные сети (IP over ATM, IP over SDH/SONET);
6. по спектру оказываемых услуг:
   - моносервисные сети (передача данных, передача голоса);
   - мультисервисные сети;
7. по типу передаваемой информации:
   - сети передачи данных;
   - сети передачи голоса;
   - сети передачи видео;
8. по наличию сигнализации:
   - сети с выделенной сигнализацией (SS7);
   - сети без выделенной сигнализации (TCP/IP);
9. по топологии сети:
   - сети с топологией шина;
   - сети с топологией кольцо;
   - сети с топологией звезда;
   - сети со смешанной топологией;
10. по среде передачи:
    - проводные сети:

[1]Хотя стандарт DECT и рассматривается как более защищённый относительно аналоговых стандартов, реализация функций защиты выполнена на крайне низком уровне. Например, используется слабое шифрование; отсутствует взаимная аутентификация трубки и базы.

  - связь осуществляется по медному кабелю;
  - связь осуществляется по оптоволокну;
- беспроводные сети.

Сеть, размещённая на ограниченной территории, например, в отдельном здании, считается локальной вычислительной сетью (ЛВС, LAN). Локальные сети характеризуются высокой скоростью передачи данных на сравнительно небольшие расстояния. На данный момент локальные сети строятся в основном на базе технологии Ethernet и её модификаций (Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet).

Совокупность нескольких локальных сетей, объединённых линиями связи, называют составной, распределённой или *глобальной сетью (Internetwork, Internet)*. Такие сети могут состоять из *подсетей (subnet)* различных технологий. Сети крупных фирм (Intranet) могут использовать технологии как локальных, так и глобальных сетей. Таким образом, глобальные вычислительные сети (WAN) характеризуются тем, что пользователи этих сетей могут находится на большом (в географическом смысле) расстоянии друг от друга, например, в разных городах, но при этом совместно использовать информационные данные.

Глобальные сети передачи цифровых данных строят на основе различных сетевых технологий, в том числе сетей с коммутацией каналов и с коммутацией пакетов.

Сети с коммутацией пакетов базируются на протоколе IP *(Internet Protocol)*, использующим дейтаграммный метод передачи сообщений. С IP-сетями совместимы сети на базе *протокола коммутации по меткам (Multi Protocol Label Switching, MPLS)*. В настоящее время MPLS рассматривается в качестве основной транспортной технологии для сетей с пакетной коммутацией. Технология *Ethernet операторского класса для глобальных сетей (Carrier Ethernet Transport, CET)* пока не получила широкого распространения, однако имеет хорошие перспективы широкого внедрения. Сети, использующие технологии виртуальных каналов (X.25; сети трансляции кадров (Frame Relay, FR); сети асинхронной передачи данных (Asynchronous Transfer Mode, ATM)), постепенно вытесняются.

Глобальные сети с коммутацией каналов используют технологии *синхронной цифровой иерархии (Synchronous Digital Hierarchy, SDH)*, *плезиохронной цифровой иерархии (Plesiochronous Digital Hierarchy, PDH)*, а также технологии *оптических линий связи спектрального уплотнения по длине волны (Wave-length Division Multiplexing, WDM)*. В настоящее время внедряются технологии *оптических транспортных сетей — ОТС (Optical Transport Network, OTN)*, объединивших технологии систем цифровой иерархии SDH и спектрального уплотнения по длине волны WDM.

Технологии PDH, SDH характеризуются высокой скоростью передачи данных. По сетям PDH передача данных осуществляется со скоростью от 2 Мбит/с до 139 Мбит/с, а в сетях SDH — от 155 Мбит/с до 40 Гбит/с. В системах со спектральным уплотнением по длине волны (технологии CWDM, DWDM) на волоконно-оптических кабелях данные передаются с ещё большей скоростью. Основными аппаратными средствами высокоскоростных технологий с коммутируемыми цифровыми линиями связи являются мультиплексоры (MUX).

В сетях с коммутацией пакетов в зависимости от предъявляемых требований могут использоваться технологии виртуальных каналов, применяемые в сетях ATM, Frame Relay или технологии передачи дейтаграммных сообщений — сети IP-технологий. В сетях с виртуальными каналами предварительно прокладывается маршрут, по которому передаются данные. После приёма данных адресат подтверждает их получение. Это обеспечивает надёжность передачи.

Технология X.25 использует ненадёжные аналоговые линии связи. Её характерной чертой является низкая скорость передачи данных (до 48 кбит/с). Тем не менее, X.25 используется, например, в сетях банкоматов, поскольку обеспечивает высокую надёжность передачи данных при ненадёжных линиях.

Технология Frame Relay обеспечивает более высокую по сравнению с X.25 скорость передачи данных до 2–4 Мбит/с. Но линии связи должны быть более надёжными по сравнению с X.25. Более высокую скорость передачи данных (155 или 620 Мбит/с, а также 2,4 Гбит/с) обеспечивают сети ATM. Однако развитие этих сетей сдерживает их высокая стоимость.

Компромиссное решение по цене и скорости передачи данных предоставляют IP-сети, получившие в настоящее время наиболее широкое распространение. Поэтому на базе IP-сетей создаётся транспортный уровень мультисервисных сетей NGN с распределённой коммутацией пакетов.

Ещё одна технология — технология *виртуальных частных сетей (Virtual Private Network, VPN)* использует для передачи данных сеть общего пользования Интернет, формируя защищённые каналы связи с гарантированной полосой пропускания и требуемым уровнем безопасности передаваемой информации. Таким образом, сети VPN при экономичности и доступности обеспечивают безопасность и секретность передаваемых сообщений. Использование VPN позволяет сотрудникам фирмы получить безопасный дистанционный доступ к корпоративной сети компании через Интернет.

1.6. Стандартизирующие организации

Организации в международной системе стандартизации можно разделить следующим образом [1; 2]:

- официальные международные организации стандартизации:
  - *Международная организация по стандартизации (International Organization for Standardization, ISO)*
    Создана в 1946 г., включает в себя национальные организации стандартизации из 157 стран мира, в частности, ANSI (США), Федеральное агентство по техническому регулированию и метрологии (Россия), BSI (Великобритания), AFNOR (Франция) и др., обладает полномочиями для координирования на международном уровне разработки различных промышленных стандартов и принятия их в качестве международных стандартов.
  - *Международный союз электросвязи, МСЭ (International Telecommunication Union, ITU*[1]*)*
    Занимается стандартизацией международных средств связи и состоит из трёх основных секторов:
    - сектор стандартизации телекоммуникаций (ITU-T[2]) — занимается вопросами, связанными с телефонными системами и системами передачи данных[3];
    - сектор радиосвязи (ITU-R) — распределяет радиочастоты между конкурирующими компаниями, решает спорные вопросы в данной области;
    - сектор развития (ITU-D) — занимается вопросами стратегии и политики развития систем электросвязи;
- региональные организации стандартизации:
  - *Европейский институт стандартизации в области телекоммуникаций (European Telecommunications Standards Institute, ETSI)*
    Создан в 1988 г. Отвечает за стандартизацию информационных и телекоммуникационных технологий в пределах Европы.
  - *Центр сетевых информационных технологий Азиатско-Тихоокеанского региона (Asia Pacific Network Information Centre, APNIC)*
    Отвечает за распределение сетевых ресурсов в Азиатско-тихоокеанском регионе;
- национальные организации стандартизации:
  - *Федеральное агентство по техническому регулированию и метрологии (Россия)*;

[1] ITU также принято переводить как Международный телекоммуникационный союз.

[2] С 1956 по 1993 г. ITU-T именовался CCITT (Comité Consultatif International Télégraphique et Téléphonique) — Консультативный комитет по международной телефонной и телеграфной связи.

[3] Рекомендации ITU-T часто становятся международными стандартами, хотя любая страна может и проигнорировать их.

  - *Американский институт национальных стандартов (American National Standards Institute, ANSI)*;
  - и др.;
- промышленные консорциумы:
  - *Сообщество инженеров по электротехнике и электронике (Institute of Electrical and Electronic Engineers, IEEE)*
    Целью данной организации является продвижение теоретических и прикладных достижений электротехнической и электронной индустрии.
  - *Рабочая группа по проектированию Интернет-технологий (Internet Engineering Task Force, IETF)*
    IETF представляет собой сообщество разработчиков, операторов, изготовителей и исследователей в области сетевых технологий. В основе Интернет-стандартизации лежит технология издания и поддержания RFC-документов — спецификаций, разработанных различными организациями и рабочими группами IETF.
  - *Интернет-сообщество (Internet Society, ISOC)*
    ISOC представляет собой ассоциацию экспертов, отвечающих за разработку стандартов технологий сети Интернет.
  - *Консорциум, специализирующийся в области разработки и развития стандартов WWW-технологий (World Wide Web Consortium, W3C).*

1.7. Краткие итоги раздела

1. Телекоммуникационная сеть образуется совокупностью конечных узлов и сетевых элементов, соединённых линиями (каналами) связи.
2. Аппаратура абонентов называется конечными узлами (хостами, Host).
3. Соединение узлов (абонентов) между собой производится через транзитные — промежуточные сетевые элементы (СЭ).
4. Различают сети с коммутацией каналов, когда сетевые элементы выполняют функции коммутаторов, и с коммутацией пакетов (сообщений), когда сетевые элементы выполняют роль маршрутизаторов.
5. Архитектура сетей связи классифицируется согласно требованиям Единой сети электросвязи Российской Федерации (ЕСЭ РФ).
6. Терминальным оборудованием сети являются компьютеры, телефонные аппараты, теле- или радиоприёмники.
7. Локальные сети объединяют пользователей в пределах комнаты, здания или группы близко стоящих зданий (кампуса). Локальные сети строятся на базе протокола Ethernet и его модификаций (Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet).
8. Магистральная сеть объединяет отдельные сети доступа, обеспечивая транспорт трафика между ними по высокоскоростным каналам.

9. Для создания маршрута в разветвлённой сети необходимо задавать адреса источника и получателя сообщения.
10. Сети передачи данных с коммутацией пакетов подразделяются на локальные и глобальные.
11. В создаваемых в настоящее время мультисервисных сетях следующего поколения (NGN) технология коммутации пакетов используется для передачи всех видов трафика.
12. Различают виртуальные и дейтаграммные соединения или сети.
13. Технологии виртуальных каналов применяются в сетях ATM, Frame Relay.
14. Сети технологии IP являются дейтаграммными, когда отсутствует предварительное соединение конечных узлов и нет подтверждения приёма сообщения.
15. Высокую надёжность обеспечивает протокол управления передачей TCP.

1.8. Вопросы по разделу

1. Что собой представляют телекоммуникационные сети?
2. Чем отличаются сети с коммутацией каналов от сетей с коммутацией сообщений (пакетов)?
3. Какие функции выполняет маршрутизатор?
4. Что собой представляет метрика протокола маршрутизации?
5. Чем отличается коммутации пакетов от коммутации сообщений?
6. Что содержит служебная информация пакетов?
7. Чем отличаются локальные и глобальные сети передачи данных?
8. Чем отличаются виртуальные и дейтаграммные соединения?
9. Какой протокол обеспечивает надёжность передачи данных?
10. Какие технологии (коммутации каналов или коммутации пакетов) используются в мультисервисных сетях следующего поколения (NGN) для передачи всех видов трафика? Почему?
11. Что такое протокол?
12. Что такое интерфейс?
13. Дайте определение следующим понятиям: сеть связи, линия связи, технология коммутации, протокол, услуга, интерфейс.
14. Приведите классификацию сетей телекоммуникаций.
15. Укажите основные стандартизирующие организации в сфере телекоммуникаций, охарактеризуйте их деятельность.

Глава 2. Принципы построения телекоммуникационных сетей

В главе раскрывается понятие эталонной модели открытых систем (OSI), даётся подробная характеристика каждого её уровня, рассматривается иерархия протоколов в различных стеках (OSI, TCP/IP, IEE802, IPX/SPX, NetBIOS/SMB, H.323, SS7) и их соотношение с уровнями модели OSI, даётся краткое описание некоторых протоколов рассматриваемых стеков. В основу главы положены материалы из источников [1—12].

В результате освоения данной темы студент должен:

знать:

- эталонную модель взаимодействия открытых систем;
- модели TCP/IP, IEEE802;
- перечень основных протоколов различных стеков сетей и систем передачи информации;

уметь:

- зная основные функции протоколов различных стеков, определять их место в модели взаимодействия открытых систем;

владеть:

- навыками анализа основных характеристик уровней модели взаимодействия открытых систем;
- навыками анализа основных характеристик протоколов различных стеков.

2.1. Обзор эталонной модели OSI

Сложность сетевых структур и разнообразие телекоммуникационных устройств, выпускаемых различными фирмами, привели к необходимости стандартизации как устройств, так и процедур обмена данными между пользователями. В начале 1980-х гг. при содействии ряда международных организаций, в частности Международной организации по стандартизации (International Standards Organization, ISO) была создана базовая эталонная модель взаимодействия открытых систем (Open System Interconnection reference model, OSI), которая определила концепцию и методологию создания сетей передачи данных.

Модель описывает стандартные правила функционирования и взаимодействия устройств и программных средств при обмене данными между узлами в открытой системе. Открытая система состоит из программно-аппаратных средств, способных взаимодействовать между собой при использовании стандартных правил и интерфейсов.

Модель OSI имеет иерархическую структуру и чётко определяет семь уровней взаимодействия систем, стандартизует имена уровней и указывает услуги и функции каждого уровня (рис. 2.1).

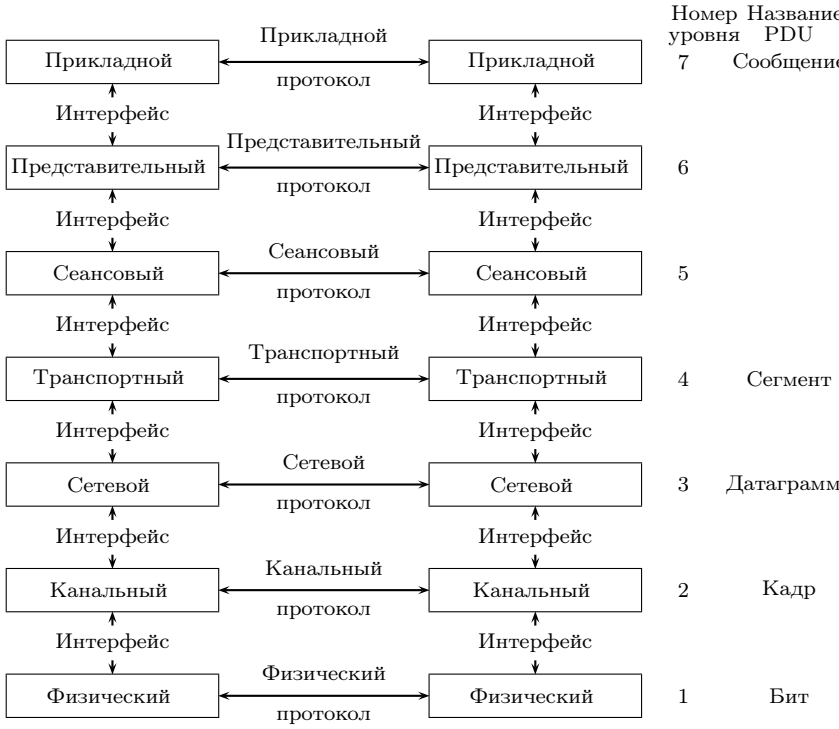


Рис. 2.1. Эталонная модель OSI [1; 2]

Как было сказано в разделе 1.2, нижележащий уровень посредством *интерфейса* предоставляет некоторый набор услуг (сервисов) для вышележащего.

Одноранговые сущности модели обмениваются определёнными фрагментами данных (Protocol Data Unit, PDU):

– на физическом уровне — последовательность битов;
– на канальном уровне — кадры (Frame);
– на сетевом уровне — пакеты (Packet);
– на транспортном уровне — сегменты (Segment);
– на трёх верхних уровнях — сообщения (Data).

При взаимодействии двух узлов сети данные последовательно проходят на первом узле от уровня приложений до физического уровня, затем передаются по физической среде на физический уровень второго узла и постепенно доходят до его уровня приложений. При этом в процессе прохождения уровней на узле происходит инкапсуляция (деинкапсулация) передаваемых PDU (рис. 2.2).

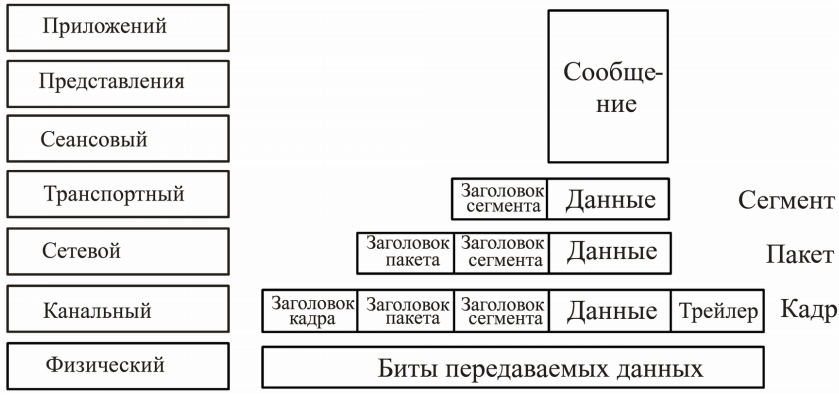


Рис. 2.2. Инкапсуляция данных [3]

Передаваемое сообщение, сформированное приложением, проходит три верхних сетенезависимых уровня и поступает на транспортный уровень, где делится на части и каждая часть инкапсулируется (помещается) в сегмент данных (рис. 2.2). В заголовке сегмента содержится идентификатор протокола верхнего уровня, с помощью которого подготовлено сообщение, и идентификатор протокола, который будет обрабатывать данный сегмент.

На сетевом уровне сегмент инкапсулируется в пакет данных, в заголовке (header) которого содержатся в том числе и сетевые адреса отправителя (Source Address, SA) и получателя (Destination Address, DA).

На канальном уровне пакет инкапсулируется в кадр, в заголовке которого, в частности, содержатся физические адреса узла передатчика и приёмника. Кроме того, на этом уровне добавляется *трейлер* (концевик) кадра, содержащий информацию, необходимую для проверки правильности принятой информации. Таким образом, происходит обрамление данных заголовками со служебной информацией, т.е. инкапсуляция данных.

2.1.1. Принципы построения модели OSI

Эталонная модель OSI базируется на следующих принципах:

1. уровень должен создаваться по мере необходимости выделения отдельного уровня абстракции;
2. каждый уровень должен выполнять строго определённую функцию;
3. функции для каждого уровня должны выбираться с учётом создания стандартизованных международных протоколов;
4. границы между уровнями должны выбираться так, чтобы поток данных между интерфейсами был минимальным;
5. количество уровней должно быть достаточно большим, чтобы различные функции не объединялись в одном уровне без необходимости, но не слишком высоким, чтобы архитектура не становилась громоздкой.

2.1.2. Уровни в модели OSI

Одним из важнейших принципов OSI является то, что сетевые системы взаимодействуют друг с другом на одинаковых уровнях модели. Дадим краткое описание уровней модели OSI (рис. 2.1).

Уровень 1: Физический уровень

Физический уровень (Physical Layer) обеспечивает передачу битовых потоков без каких-либо изменений между логическими объектами уровня звена данных по физическим соединениям.

На данном уровне определяются базовые механизмы кодирования и декодирования двоичных данных в физическом носителе, а также специфицируются соединители, но не сама среда. Среда, согласно эталонной модели, рассматривается как нечто, лежащее ниже физического уровня. Битовый поток в носителе должен быть независим от типа среды.

Физический уровень предоставляет канальному уровню следующие услуги и элементы услуг:

– физические соединения;
– физические сервисные блоки данных;
– физические оконечные пункты соединения;
– осуществляет идентификацию канала данных;
– осуществляет упорядочение;
– осуществляет оповещение об ошибках;
– определяет параметры качества услуги.

На физическом уровне выполняются следующие функции:

– активизация и деактивизация физического соединения;
– передача физических сервисных блоков данных;
– административное управление физическим уровнем.

Уровень 2: Канальный уровень

Канальный уровень (Data Link Layer) также носит названия *уровень управления передачей данных (Data Link Control, DLC)* или *уровень звена данных.*

Канальный уровень обеспечивает функциональные и процедурные средства для установления, поддержания и разрыва соединений канального уровня между сетевыми логическими объектами и для передачи сервисных блоков данных этого уровня. Соединение канального уровня строится на основе одного или нескольких физических соединений.

Канальный уровень обнаруживает и по возможности исправляет ошибки, которые могут возникнуть на физическом уровне. Кроме того, канальный уровень обеспечивает для сетевого уровня возможность управлять подключением каналов данных на физическом уровне. Единицу информации на канальном уровне называют *кадром (Frame).*

Канальный уровень предоставляет следующие услуги или элементы услуг сетевому уровню:

- соединение канального уровня;
- сервисные блоки данных канального уровня;
- идентификаторы оконечного пункта соединения канального уровня;
- осуществляет упорядочение блоков данных;
- осуществляет оповещение об ошибках;
- управляет потоком данных;
- определяет параметры качества услуги.

На канальном уровне выполняются следующие функции:

- установление и разрыв соединения канального уровня;
- отображение сервисных блоков данных канального уровня;
- расщепление соединения канального уровня;
- разграничение и синхронизация;
- упорядочение блоков данных;
- обнаружение ошибок;
- восстановление при ошибках;
- управление потоком данных;
- идентификация и обмен параметрами;
- управление переключением каналов данных;
- административное управление канальным уровнем.

Уровень 3: Сетевой уровень

Сетевой уровень (Network Layer) предоставляет средства установления, поддержания и разрыва сетевого соединения, а также функциональные и процедурные средства для обмена по сетевому соединению сетевыми сервисными блоками данных между транспортными логическими объектами.

Сетевой уровень обеспечивает транспортным логическим объектам независимость от функций маршрутизации и ретрансляции, связанных с процессами установления и функционирования данного сетевого соединения.

Все функции ретрансляции и расширенные протоколы последовательного переноса данных, которые предназначены для поддержания сетевых услуг между оконечными открытыми системами, функционируют ниже транспортного уровня. Единицу информации на сетевом уровне называют *датаграммой или дейтаграммой (Datagram)*.

Основной услугой сетевого уровня является обеспечение передачи данных без каких-либо изменений между транспортными логическими объектами, т.е. структура и содержание данных, предоставляемых для передачи, определяется уровнями, расположенными выше сетевого.

Услуги, предоставляемые на каждом из концов сетевого соединения, одинаковы и в том случае, когда сетевое соединение проходит через несколько подсетей, каждая из которых предоставляет различные услуги.

Сетевой уровень предоставляет следующие услуги:
- сетевые адреса;
- сетевые соединения;
- сетевые идентификаторы оконечных пунктов соединения;
- осуществляет передачу сетевых сервисных блоков данных;
- определяет параметры качества услуги;
- оповещает об ошибках;
- упорядочивает блоки данных;
- управляет потоком данных;
- осуществляет передачу срочных сетевых сервисных блоков данных;
- осуществляет сброс;
- осуществляет разрыв сетевого соединения.

Некоторые из этих услуг являются необязательными:
- пользователь должен запросить услугу;
- поставщик сетевой услуги может удовлетворить запрос или сообщить, что запрошенная услуга недоступна.

Функции сетевого уровня обеспечивают использование различных конфигураций для поддержки сетевых соединений: от соединений, поддерживаемых двухпунктовыми сетевыми конфигурациями, до сетевых соединений, поддерживаемых сочетаниями подсетей с различными характеристиками.

Сетевой уровень выполняет следующие функции:
- маршрутизацию и ретрансляцию;
- организацию сетевых соединений;
- мультиплексирование сетевого соединения;
- сегментирование и объединение;
- обнаружение ошибок;
- восстановление при ошибках;
- упорядочение блоков данных;

– управление потоком данных;
– передачу срочных данных;
– сброс;
– выбор услуги;
– административное управление сетевым уровнем.

Уровень 4: Транспортный уровень

Транспортный уровень (Transport Layer) обеспечивает передачу данных без каких-либо изменений между сеансовыми логическими объектами и освобождает их от выполнения операций, обеспечивающих надёжную и экономически эффективную передачу данных.

Транспортный уровень оптимизирует использование доступных сетевых услуг, чтобы обеспечить пропускную способность, требуемую каждым сеансовым логическим объектом, при минимальных затратах. Эта оптимизация достигается путём внесения ограничений, обусловленных совместными требованиями со стороны всех одновременно работающих сеансовых логических объектов, а также общим качеством и объёмом сетевых услуг, предоставляемых транспортному уровню.

Все протоколы, определённые на транспортном уровне, имеют межоконечный характер. Под окончаниями понимают связанные транспортные логические объекты. Поскольку сетевые услуги обеспечивают сетевые соединения между транспортными логическими объектами по принципу «каждый с каждым», включая использование последовательно соединённых подсетей, то транспортный уровень освобождается от функций маршрутизации и ретрансляции.

На транспортном уровне имеются функции, обеспечивающие требуемое качество услуг на основе услуг, предоставляемых сетевым уровнем. Качество сетевых услуг зависит от того, как они реализуются.

Транспортный уровень однозначно идентифицирует каждый сеансовый логический объект с помощью транспортного адреса. Транспортные услуги предоставляют средства для установления, поддержания и разрыва транспортного соединения. Транспортное соединение обеспечивает дуплексную передачу между двумя транспортными адресами.

Для одной пары транспортных адресов может быть установлено несколько транспортных соединений. Сеансовые логические объекты используют идентификаторы оконечных пунктов транспортных соединений, обеспечиваемые транспортным уровнем для распознавания этих пунктов.

Качество услуг при предоставлении транспортного соединения зависит от класса обслуживания, запрашиваемого сеансовым логическим объектом при установлении транспортного соединения. Выбранное качество обслуживания поддерживается в течение существования транспортного соединения.

Транспортным уровнем предоставляются следующие виды услуг:

– установление транспортного соединения;
– передача данных;
– разрыв транспортного соединения.

На транспортном уровне могут быть реализованы следующие функции:

– преобразование транспортного адреса в сетевой;
– межоконечное мультиплексирование транспортных соединений в сетевые;
– установление и разрыв транспортных соединений;
– межоконечное упорядочение блоков данных по отдельным соединениям;
– межоконечное обнаружение ошибок и необходимый контроль за качеством услуг;
– межоконечное восстановление после ошибок;
– межоконечное сегментирование, объединение и сцепление;
– межоконечное управление потоком данных по отдельным соединениям;
– супервизорные функции;
– передача срочных транспортных сервисных блоков данных.

Уровень 5: Сеансовый уровень

Сеансовый уровень (Session Layer) реализует службу имён (отображение логических имён в сетевые адреса), устанавливает сеансы между службами и создаёт точки для контрольной синхронизации в случае потери связи.

Сеансовый уровень выполняет следующие функции:

– отображение сеансового соединения на транспортное соединение;
– управление потоком данных в сеансовом соединении;
– передачу срочных данных;
– восстановление сеансового соединения;
– административное управление сеансовым уровнем.

Уровень 6: Уровень представления

Уровень представления (Presentational Layer) устанавливает способы представления информации, которой обмениваются прикладные логические объекты или на которую они ссылаются в процессе этого обмена.

Уровень представления охватывает два взаимодополняющих аспекта способов представления информации:

– представление данных, подлежащих передаче между прикладными логическими объектами;

– представление структуры данных, которую прикладные логические объекты намереваются использовать в своём диалоге, наряду с представлениями совокупности действий, которые могут быть выполнены над этой структурой данных.

На этом уровне определяется общий синтаксис (способы представления данных), но не семантика, которая известна только прикладным логическим объектам.

Уровень представления обеспечивает способы представления информации, которые являются общими для взаимодействующих прикладных логических объектов. Таким образом, прикладные логические объекты освобождаются от функции представления информации, поскольку используется общий способ представления, и для них обеспечивается синтаксическая независимость. Такая независимость может быть реализована двумя путями.

1. На уровне представления обеспечиваются элементы поддержки синтаксиса, являющиеся общими для использующих их прикладных логических объектов.
2. Прикладные логические объекты могут использовать произвольный синтаксис, а уровень представления обеспечивает преобразование этих синтаксисов. Для обмена между прикладными логическими объектами применяется общий синтаксис. Такое преобразование выполняется внутри открытой системы. На другие открытые системы это не влияет и, следовательно, не оказывает влияние на стандартизацию протоколов уровня представления.

Уровень представления обеспечивает сеансовые услуги и добавляет к ним следующие возможности:

– преобразование синтаксиса;
– выбор синтаксиса.

Преобразование синтаксиса связано с преобразованием кодовых и символьных наборов, с модификацией расположения данных и с адаптацией действий над структурами данных. Выбор синтаксиса предоставляет средства первоначального выбора синтаксиса и последующего изменения сделанного выбора.

Прикладным логическим объектам предоставляются услуги сеансового уровня в виде услуг представления. На уровне представления выполняются следующие функции, с помощью которых реализуются услуги представления:

– запрос на установление сеанса;
– передача данных;
– соглашение по выбору и повторному выбору синтаксиса;
– преобразование синтаксиса, включая преобразование данных, форматирование и специальные функции преобразования;
– запрос на завершение сеанса.

Уровень 7: Прикладной уровень

Прикладной уровень (Application Layer) является наивысшим уровнем в эталонной модели OSI и поэтому не имеет интерфейса с более высоким уровнем. Он является единственным средством доступа прикладных процессов к функциональной среде OSI.

Прикладной уровень поддерживает локальные операционные системы, предоставляя им набор разнообразных протоколов, с помощью которых производится доступ к сетевым ресурсам. Единицу информации на прикладном уровне называют *сообщением (Message)*.

Прикладные процессы обмениваются информацией с помощью прикладных логических объектов, прикладных протоколов и услуг уровня представления.

Прикладные услуги отличаются от услуг, предоставляемых другими уровнями, тем, что они не предоставляются какому-либо верхнему уровню и не связаны ни с каким пунктом доступа к услугам. Кроме передачи информации может предоставляться следующий набор услуг:

- идентификация партнёров, собирающихся инициировать связь;
- установление уровня авторизации для взаимодействия;
- авторизация партнёров, собирающихся инициировать взаимосвязь;
- определение параметров качества услуг, считающихся приемлемыми;
- идентификация ограничений на синтаксис данных;
- и другие.

На прикладном уровне выполняются все функции связи между открытыми системами, которые не выполняются нижележащими уровнями. В их число включаются функции, выполняемые программными средствами, и функции, выполняемые людьми. По сути прикладной уровень модели OSI обеспечивает сопряжение человека с сетевыми технологиями, что позволяет пользователям общаться между собой через сеть, т.е. создаёт интерфейс между приложениями конечных устройств при передаче сообщений по сети.

2.2. Иерархия протоколов в различных стеках

2.2.1. Стек ISO/OSI

В данном случае эталонная модель первична, а стек протоколов вторичен. Это привело к некоторой тяжеловесности протоколов данного стека (рис. 2.3). Интересно, что большое количество протоколов стека разработано под влиянием IBM.

Из-за ограничений определения *физического уровня* в эталонной модели протоколы физического уровня в данном стеке практически отсутствуют (за исключением семейства протоколов X.25, которое,

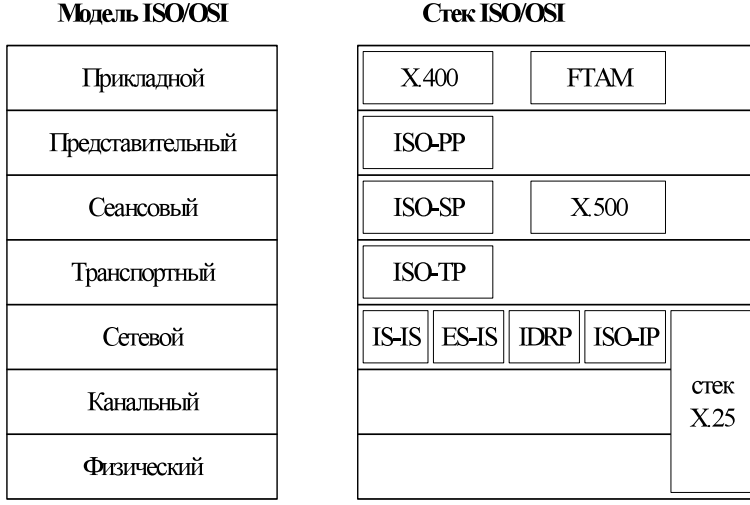


Рис. 2.3. Некоторые протоколы стека ISO/OSI

впрочем, по генезису выбивается из общего построения стека протоколов ISO/OSI).

К *канальному уровню* можно отнести протокол Logical Link Control (LLC), который хотя и разработан в рамках IEEE 802.2, но служит для сопряжения стека протоколов ISO/OSI с канальным уровнем других стеков.

Основным протоколом *сетевого уровня* является *протокол межсетевого взаимодействия ISO (ISO Internetworking Protocol, ISO-IP)*, описанный в RFC 1575[1] и документах ISO S 8473, IS 8348. Другое его название — *услуга организации сетевого взаимодействия без установления соединения (Connectionless Network Service, CLNS)*.

Функцию маршрутизации обеспечивают протоколы *IS-IS (Intermediate System to Intermediate System)* (ISO 10589), *ES-IS (End System to Intermediate System)* (ISO 9542) и *CLNS* (ISO 8473), а также *внутридоменный протокол маршрутизации (Inter Domain Routing Protocol, IDRP)* (ISO 7498).

На *транспортном уровне* располагается *транспортный протокол ISO (ISO Transport Protocol, ISO-TP)* (ISO 8073).

[1] *Hares S., Wittbrodt C.* An Echo Function for CLNP (ISO 8473), RFC 1575. URL: `http://www.faqs.org/rfcs/rfc1575.html`.

Основным протоколом *сеансового уровня* является протокол *ISO-SP (OSI Session Layer Protocol)* (соответствует спецификации ISO/IEC 8327-1 09-1996 ITU-T X.225). На этом же уровне находится *протокол доступа к каталогам X.500* (прародитель *облегчённого протокола доступа к каталогам (Lightweight Directory Access Protocol, LDAP)* стека TCP/IP). Кроме того, следует отметить *протокол ISO NetBIOS* (соответствует протоколу NetBIOS (Network Basic Input/Output System) одноимённого стека протоколов).

На уровне представления находится *протокол представления (Presentation Protocol, PP)* (ISO IS 8823).

На *прикладном уровне* присутствует набор протоколов, достаточный для основных пользовательских приложений. Здесь же следует упомянуть почтовые протоколы X.400, базирующиеся на рекомендациях CCITT с X.400 по X.430. Стандарт X.400 описывает функционирование *агентов передачи почты (Message Transfer Agents, MTA)*.

Доступ к файлам описывается *протоколом управления доступом и передачей файлов (File Transfer Access and Management, FTAM)* (аналог *протокола передачи файлов (File Transfer Protocol, FTP)* в стеке TCP/IP). Кроме того, на этом уровне находится сетевой *протокол разделения файлов (Server Message Block, SMB)*.

2.2.2. Стек TCP/IP

Эталонная модель TCP/IP документирует дизайн семейства протоколов TCP/IP и состоит из четырёх уровней (рис. 2.4, 2.5).

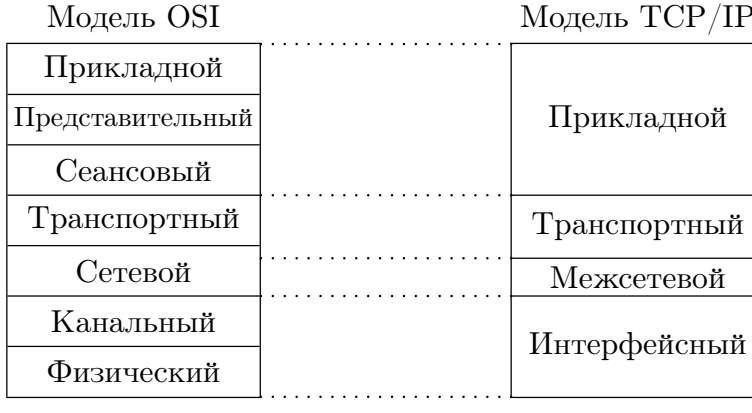


Рис. 2.4. Соответствие моделей OSI и TCP/IP

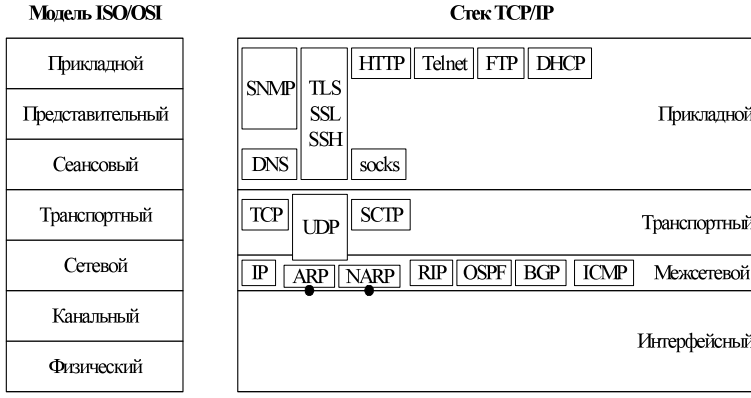


Рис. 2.5. Некоторые протоколы стека TCP/IP

Основой модели служит *межсетевой уровень*. Его задачей является доставка пакетов в пункт назначения. Передача осуществляется без установления соединения. Здесь же осуществляется выбор маршрута пакета. Пакеты могут двигаться к пункту назначения разными маршрутами, поэтому и прибывать они могут не в том порядке, в котором были отправлены.

На межсетевом уровне определён протокол *IP (Internet Protocol)*, задающий в том числе и схему адресации. Кроме того, здесь же определены протоколы маршрутизации *RIP (Routing Information Protocol)*, *OSPF (Open Shortest Path First)*, *BGP (Border Gateway Protocol)*. Таким образом, этот уровень близок сетевому уровню эталонной модели OSI.

На *транспортном уровне* модели TCP/IP решается задача поддержания связи между отправителем и получателем. Этот уровень в основном соответствует транспортному уровню эталонной модели OSI. На нём определены протоколы *TCP (Transmission Control Protocol)*, *UDP (User Datagram Protocol)*, *DCCP (Datagram Congestion Control Protocol)*, *SCTP (Stream Control Transmission Protocol)*.

Прикладной уровень объединяет все службы, представляемые системой пользовательским приложениям. В модели TCP/IP не выделяются отдельно *сеансовый* и *представительный* уровни. Отдельные их функции выполняются различными протоколами прикладного уровня. На этом уровне определены, например, почтовые протоколы *SMTP (Simple Mail Transfer Protocol)*, *IMAP4 (Internet Message Access Protocol rev 4)*, *POP3 (Post Office Protocol version 3)*, протокол передачи гипертекста *HTTP (Hypertext Transfer Protocol)*, протокол передачи файлов *FTP (File Transfer Protocol)*, протокол эмуляции терминала *Telnet* и др.

Интерфейсный уровень отвечает за взаимодействие между компьютером и физическим сетевым оборудованием. Он приблизительно соответствует канальному и физическому уровням модели OSI. Интерфейсный уровень по-настоящему не описан в документации по архитектуре TCP/IP, в которой сказано только, что он обеспечивает доступ к сетевой аппаратуре системно-зависимым способом.

2.2.3. Стек IEEE 802

Семейство протоколов IEEE 802 базируется на фирменных стандартах построения локальных сетей Arcnet, Ethernet, Token Ring.

Протоколы IEEE 802 охватывают только два нижних уровня семиуровневой эталонной модели OSI, а именно физический и канальный (рис. 2.6). Это связано с тем, что именно эти уровни в наибольшей степени отражают специфику локальных сетей.

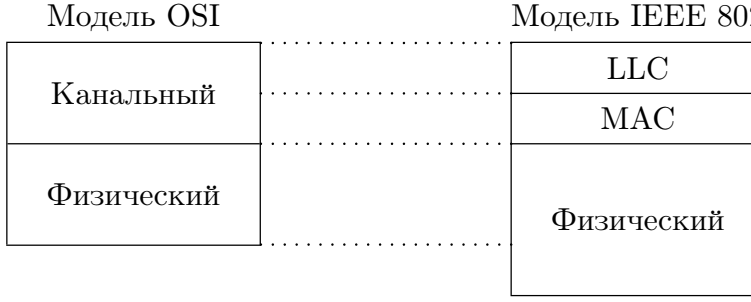


Рис. 2.6. Соответствие эталонных моделей OSI и IEEE 802

На *физическом уровне* модели IEEE 802 специфицируются также и различные типы носителей, то есть среда передачи, что не входит в определение физического уровня эталонной модели OSI. Поэтому физический уровень модели IEEE 802 изображён охватывающим область, лежащую ниже физического уровня модели OSI.

В спецификации IEEE *канальный уровень (Data Link Control, DLC)* разделяется на уровень *управления логическим каналом (Logical Linc Control, LLC)* и уровень *управления доступом к носителю (Media Access Control, MAC)*. По сути, уровень MAC эквивалентен всему уровню DLC в предыдущих спецификациях. Добавление уровня LLC является результатом давления IBM, разрабатывавшей стандарт Token Ring одновременно со спецификацией IEEE 802.5. Поэтому уровень LLC — это отражение операций *высокоуровнего протокола управления каналом передачи данных (High-Level Data Link Control, HDLC)* в *системной сетевой архитектуре (Systems Network Architecture, SNA)*.

2.2.4. Стек IPX/SPX

Стек протоколов IPX/SPX (или стек Novell NetWare) разработан в начале 1980-х гг. фирмой Novell для сетевой операционной системы NetWare.

Стек включает в себя следующие протоколы (рис. 2.7): *протокол межсетевого обмена (Interwork Packet Exchange, IPX)*, *протокол маршрутизации (Routing Information Protocol, RIPX)*[1], *протокол упорядоченного обмена пакетами (Sequenced Packet Exchange, SPX)*, *протокол анонсирования сервиса (Service Advertising Protocol, SAP)*, *протокол ядра NetWare (Netware Core Protocol, NCP)* и др.

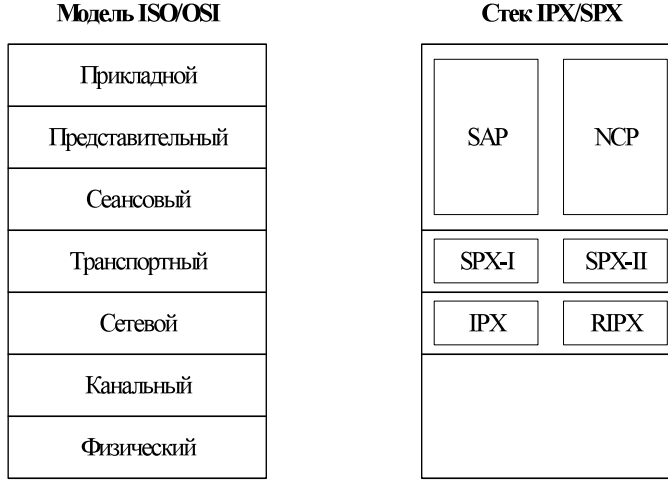


Рис. 2.7. Некоторые протоколы стека IPX/SPX

2.2.5. Стек NetBIOS/SMB

Стек NetBIOS/SMB разработан в 1984 г. совместно IBM и Microsoft для сетей IBM PC Network и IBM Token Ring.

Протокол *NetBIOS (Network Basic Input/Output System)* был разработан как аналог системы BIOS персонального компьютера. Он реализует большинство услуг и функций сетевого, транспортного и

[1] Следует различать протокол RIP в стеке TCP/IP и протокол RIPX в стеке IPX/SPX.

сеансового уровней модели ISO/OSI (так, протокол NetBIOS не поддерживает маршрутизацию пакетов, что является одной из основных функций сетевого уровня). Однако впоследствии за протоколом NetBIOS остался только сеансовый уровень, поскольку на более низких уровнях стали использовать стандартные протоколы (например, TCP/IP или IPX/SPX).

Следует отметить, что существует три реализации протокола NetBIOS:

- *NetBEUI (NetBIOS Extended User Interface)* — NetBIOS поверх LLC;
- *NBT (NetBIOS over TCP/IP)* — NetBIOS поверх IP;
- *NetBIOS* — NetBIOS поверх IPX.

Протокол *SMB (Server Message Block)* реализует услуги и функции прикладного уровня и уровня представления модели ISO/OSI. Протокол регламентирует взаимодействие рабочей станции с сервером. В его функции входит создание и разрыв логического соединения между рабочей станцией и сетевыми ресурсами файлового сервера, управление доступом к файлам на файловом сервере, управление очередью печати на сервере печати.

2.2.6. Стек H.323

Стандарт H.323 входит в серию рекомендаций H.32x ITU-T, разработанных для регламентации проведения аудио- и видеоконференций по телекоммуникационным сетям:

- H.320 регламентирует организацию мультимедийной связи по сетям ISDN;
- H.321 регламентирует организацию мультимедийной связи по сетям ATM;
- H.322 регламентирует организацию мультимедийной связи по сетям с коммутацией пакетов с гарантированной пропускной способностью;
- H.323 регламентирует организацию мультимедийной связи по сетям с коммутацией пакетов с негарантированной пропускной способностью;
- H.324 регламентирует организацию мультимедийной связи по телефонным сетям общего пользования;
- H.324/C регламентирует организацию мультимедийной связи по сетям мобильной связи.

По сути, H.323 является набором управляющих протоколов (рис. 2.8), строго регламентирующих использование программ (кодеков) и протоколов других стеков для организации мультимедийной связи по сетям с коммутацией пакетов.

За управление соединением и сигнализацией отвечают следующие протоколы:

– H.225.0 — протокол сигнализации и пакетирования мультимедийного потока;
– H.225.0/RAS — протокол, определяющий процедуры регистрации, доступа и состояния;
– H.245 — протокол управления для мультимедиа.

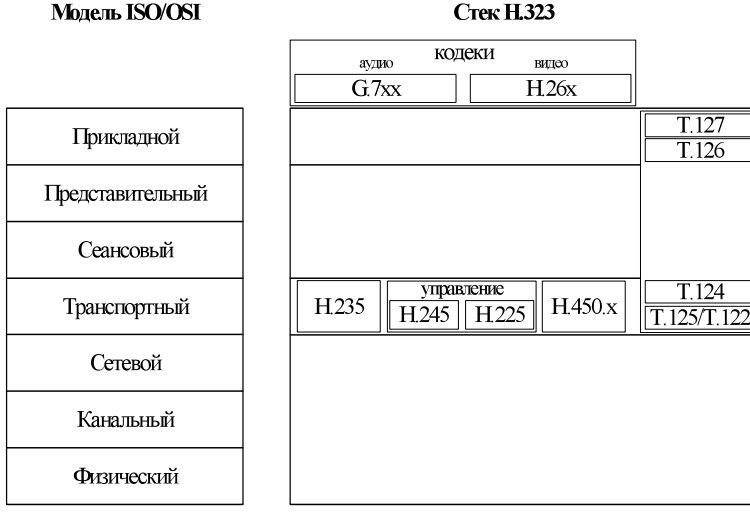


Рис. 2.8. Некоторые протоколы стека H.323

За безопасность и шифрование отвечает протокол H.235.

Протоколы H.450.x определяют различные дополнительные услуги:

– H.450.1 — определяет функции для управления дополнительными услугами;
– H.450.2 — осуществляет перевод соединения третьему абоненту;
– H.450.3 — осуществляет переадресацию вызова;
– H.450.4 — осуществляет удержание вызова;
– H.450.5 — осуществляет прикрепление вызова (park) и ответ на вызов (pick up);
– H.450.6 — осуществляет уведомление о вызове в режиме разговора;
– H.450.7 — осуществляет индексацию ожидающего сообщения;
– H.450.8 — осуществляет идентификацию имён;
– H.450.9 — осуществляет завершение соединения.

За организацию конференц-связи для передачи данных отвечает стек T.120, включающий в себя протоколы T.123, T.124, T.125.

Для обработки аудиосигнала применяются кодеки серии G.7xx: G.711, G.722, G.723.1, G.728, G.729.

Для обработки видеосигнала используются кодеки H.261, H.263, H264.

2.2.7. Стек SS7

Система сигнализации № 7 (SS7 — Signaling System 7, или ОКС7 — система общеканальной сигнализации № 7) разработана и стандартизована ITU-T в 1981 г. и представляет собой набор протоколов сигнализации, предназначенных для обмена информацией управления вызовами между коммутационными станциями и специализированными узлами сетей связи для поддержки как голосовых, так и неголосовых служб [11; 12]. SS7 образует собственную сеть, работающую параллельно цифровой сети связи.

Стек SS7 имеет четыре уровня, соответствующие физическому, канальному, сетевому и прикладному уровням модели ISO/OSI (рис. 2.9).

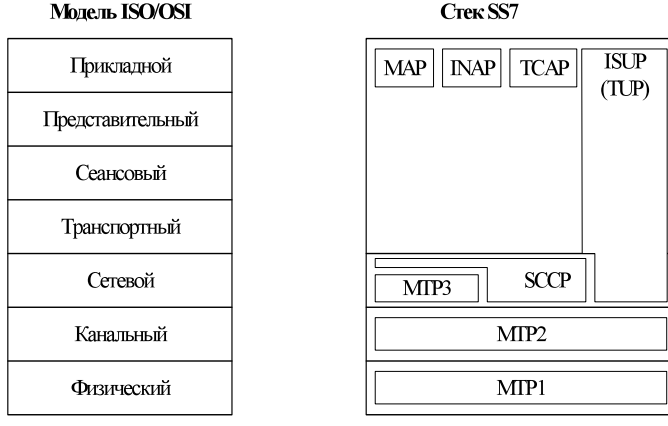


Рис. 2.9. Некоторые протоколы стека SS7

Подсистема передачи сообщений (Message Transfer Part, MTP) состоит из трёх уровней — *MTP1*, *MTP2*, *MTP3*, образующих общую транспортную подсистему, обеспечивающую корректную передачу информации между узлами сети сигнализации.

Уровень MTP1 соответствует физическому уровню модели ISO/OSI. На нём определены физические, электрические и функциональные характеристики звена данных сигнализации и средства доступа к нему.

Уровень MTP2 соответствует канальному уровню модели ISO/OSI. На нём определены функции и процедуры, относящиеся к передаче сигнальных сообщений по отдельному звену сигнализации.

На уровне MTP3 определены процедуры и функции сети сигнализации по маршрутизации сообщений, есть возможность восстановления способности передачи сигнальных сообщений после сбоев в сети, но лишь частично поддерживается адресация. Поэтому данный уровень лишь частично можно соотнести с сетевым уровнем модели ISO/OSI. Соответствие уровней становится полным, если рассматривать данный уровень совместно с *подсистемой управления соединениями сигнализации (Signaling Connection Control Part, SCCP)*.

Подсистемы MTP и SCCP в совокупности образуют *подсистему сетевых услуг (Network Service Part, NSP)*.

Протокол *ISUP (ISDN User Part)* определяет сигнальные функции для установления соединений с возможностью предоставления услуг *цифровой сети с интеграцией служб (Integrated Service Digital Network, ISDN)*. Ранее функции по управлению вызовами выполняла подсистема *TUP (Telephone User Part)*, впоследствии полностью вошедшая в ISUP. По отношению к модели ISO/OSI ISUP занимает сетевой и прикладной уровни.

На прикладном уровне модели ISO/OSI располагаются прикладная *подсистема обеспечения транзакций (Transaction Capabilities Applications Part, TCAP)*, *подсистема пользовательской мобильной связи (Mobile Application Part, MAP)* и *протокол интеллектуальной сети (Intelligent Network Application Protocol, INAP)*.

2.3. Краткие итоги раздела

1. Эталонная модель взаимодействия открытых систем OSI определяет концепцию и методологию создания сетей передачи данных и включает семь уровней.
2. Виртуальный обмен между соответствующими уровнями конечных узлов происходит определёнными единицами информации (PDU). На трёх верхних уровнях модели OSI – это сообщения, на транспортном уровне – сегменты, на сетевом уровне – пакеты, на канальном уровне – кадры и на физическом – последовательность битов.
3. Обрамление единиц информации заголовками со служебной информацией, называется инкапсуляцией.

2.4. Вопросы по разделу

1. Укажите принципы построения эталонной модели OSI.
2. Каковы основные функции уровней модели OSI?

3. Какими уровнями представлена модель TCP/IP?
4. Какими уровнями представлена модель IEEE802?
5. Что собой представляет инкапсуляция данных?
6. Опишите иерархию и назначение протоколов в различных стеках.

2.5. Примеры заданий

Задание 2.1 — Предложите возможное расширение стека TCP/IP.

Ответ (Задание 2.1) — Возможеным расширением стека TCP/IP представляется включение в стек элементов стека IEEE 802, а именно физического и канального уровней. При этом не представляется оправданным разбиение канального уровня на 2 подуровня.

2.6. Задания для самостоятельной работы

Задание 2.2 — Предложите свой вариант стека протоколов (с разделением функций и услуг по уровням).

Глава 3. Физический уровень

Глава посвящена подробному изучению физического уровня модели взаимодействия открытых систем и среды передачи данных. В ней рассматриваются основные элементы и принципы построения структурированной кабельной системы организации, приводятся основные типы модуляции сигналов и изучаются базовые механизмы кодирования и декодирования двоичных данных в физическом носителе. В основу главы легли материалы из источников [1; 2; 13—15].

В результате освоения данной темы с учётом выполнения лабораторной работы из раздела 9.1 студент должен:

знать:

- основные типы сред передачи данных, их характеристики, область их применения;
- типы модуляции аналоговых сигналов;
- основные методы кодирования сигналов;

уметь:

- определять элементы структурированной кабельной системы организации, ограничения их применения;
- строить временные диаграммы информационных сигналов;

владеть:

- способностью анализировать структурированную кабельную систему организации, определять её узкие места.

3.1. Среда передачи

Напомним, что физический уровень модели взаимодействия открытых систем отвечает за передачу потока бит между логическими объектами канального уровня, используя физические соединения. На этом уровне определяется спецификация соединителей (но не сама среда передачи), методы кодирования и декодирования двоичных данных в физическом носителе, т.е. согласно принципам иерархической структуры модели OSI сам передаваемый поток не должен зависеть от типа среды передачи.

Среда передачи может быть *кабельной* или *беспроводной*. В первом случае это может быть коаксиальный кабель (толстый или тонкий), витая пара (неэкранированная или экранированная[1]) или оптоволокно (одномодовое, многомодовое).

[1]Экран представляет собой или токопроводящую фольгу, которая блокирует высокочастотное электромагнитное излучение, или переплетённую медную проволку, которая хорошо защищает от низкочастотных наводок.

Во втором случае в качестве среды передачи выступают беспроводные радиоканалы, радиорелейная, спутниковая и сотовая связь.

Выбор среды определяется требуемой скоростью, расстоянием, надёжностью, стоимостью.

Основные характеристики среды, оказывающие влияние на передачу сигнала:

- помехи (шумы) — любой посторонний сигнал, смешивающийся с основным сигналом приёма/передачи и искажающий его;
- скорость передачи информации — количество передаваемых бит в единицу времени;
- ширина полосы пропускания — диапазон частот передатчика, в пределах которого обеспечивается передача сигнала без существенного искажения его формы (выражается в периодах в секунду, или герцах (Гц));
- пропускная способность канала — максимально возможная при определённых условиях скорость передачи данных по каналу связи;
- уровень ошибок — частота появления ошибок (под ошибкой понимается приём 1 вместо переданного 0, и наоборот).

В настоящее время среда передачи на основе медного кабеля достаточно широко используется в локальных сетях и в сетях доступа. Коаксиальные кабели применяются, главным образом, в кабельном телевидении. В локальных сетях в основном применяется витая пара и беспроводные технологии. В сетях наряду с медными кабелями широко используются волоконно-оптические кабели, основным достоинством которых является отсутствие перекрёстных и электромагнитных помех от внешних источников, что позволяет передавать сигналы с большей скоростью и на большее расстояние по сравнению с медным кабелем.

3.1.1. Коаксиальный кабель

В современных компьютерных сетях коаксиальный кабель не используется в качестве среды передачи по причине высокой стоимости и возникающих при его эксплуатации проблем. Тем не менее в качестве исторической справки данном разделе дано краткое представление о нём.

Коаксиальный кабель представляет собой электрический кабель, который состоит из двух цилиндрических проводников — внутреннего и внешнего. Внутренний медный проводник располагается в центре и изолирован пластиковым материалом. Поверх изоляционного материала располагается медная сетка или алюминиевая фольга, покрытая изолирующей внешней оболочкой. Такая конструкция позволяет успешно бороться с помехами. Однако монтаж коаксиального кабеля сравнительно сложен.

Сети, использующие толстый коаксиальный кабель, имеют определённые ограничения:

– расстояние между узлами сети должно быть кратно 2,5 м;
– в одном сегменте может быть не более 100 узлов;
– максимальная длина сегмента не должна превышать 500 м;
– сеть не должна иметь более 5 сегментов и 4 повторителей, причём только к 3 сегментам могут подключаться оконечные устройства;
– скорость передачи ограничена 10 Мбит/с;
– повреждение кабеля или соединения в любом сегменте приводит к неработоспособности сети;
– низкая устойчивость к статическому напряжению и грозовым наводкам.

Сети, использующие тонкий коаксиальный кабель, также имеют определённые ограничения:

– расстояние между устройствами не должно быть более 0,5 м;
– к одному сегменту не должно быть подключено более 30 узлов;
– максимальная длина сегмента ограничена 185 м;
– максимально число оконечных устройств пользователей в локальной сети — 90;
– для подключения сетевых адаптеров к кабелю используются специальные соединители — Т-коннекторы (T-Connector);
– скорость передачи ограничена 10 Мбит/с;
– повреждение кабеля или соединения в любом сегменте приводит к неработоспособности сети;
– низкая устойчивость к статическому напряжению и грозовым наводкам.

3.1.2. Витая пара

В локальных сетях основной средой передачи является кабель на основе витой пары, представляющей собой одну или несколько изолированных пар проводников, скрученных между собой. Скручивание пар проводов обеспечивает уменьшение взаимных наводок во время передачи сигнала (подавление перекрестных помех).

Как было сказано ранее, витая пара может быть защищена экраном. Экран из плетёной медной проволоки обеспечивает защиту от низкочастотных наводок, а из токопроводящей фольги (плёнки) — позволяет блокировать высокочастотное электромагнитное излучение.

Витая пара характеризуется *категорией (Category)* — частотным диапазоном (табл. 3.1), в котором применение типа кабеля эффективно и от которого зависит используемый вид линии связи. Категории определены в стандарте EIA/TIA 568A.

Другие характеристики витой пары, влияющие на качество передаваемого сигнала: затухание, величина перекрёстных наводок, сопротивление, ёмкость.

Виды витой пары:

Таблица 3.1

Классификация витой пары по категориям

Категория	Полоса частот, МГц	Число пар	Тип сети	Скорость
1	0,1	1	Аналоговая телефонная сеть	-
2	1	2	ISDN, Token Ring, ARCNet	до 4 Мбит/с
3	16	2 или 4	Ethernet, Token Ring	10 Мбит/с (10Base-T) или 100 Мбит/с (100Base-T4)
4	20	4	Token Ring	16 Мбит/с по одной паре (10BASE-T, 100BASE-T4)
5	100	4	Fast Ethernet	до 100 Мбит/с (100Base-TX — используются 2 пары)
5e	125	4	Gigabit Ethernet	до 1 Гбит/с (1000Base-TX — используются 4 пары)
6	200 (250)	4	Fast Ethernet, Gigabit Ethernet	до 10 Гбит/с
6a	500	4	Gigabit Ethernet	до 10 Гбит/с
7	600–700	4	-	до 10 Гбит/с
7a	1000–1200	4	-	40–100 Гбит/с

– неэкранированная (Unshielded Twisted Pair, UTP) — нет защитного экрана;
– фольгированная (Foiled Twisted Pair, FTP или F/UTP) — имеет общий экран из фольги;
– защищённая (Shielded Twisted Pair, STP) — имеет экран для каждой пары проводников и общий сеточный экран;
– фольгированная экранированная (Screened Foiled Twisted Pair, S/FTP или SSTP) — имеет экран для каждой пары проводников и общий внешний экран из медной оплётки;
– незащищенный кабель с экранированием (Unshielded Screened Twisted Pair, U/STP) — каждая пара имеет защиту из фольги, но нет общего экрана;
– экранированная с защитой (Screened Foiled Unshielded Twisted Pair, SF/UTP или SFTP) — имеется два внешних экрана (из медной сетки и фольги) с дренажным проводом между ними.

3.1.3. Оптоволокно

Волоконно-оптический кабель состоит нескольких оптоволокон, окружённых общей защитной оболочкой. Оптоволокно представляет собой двухслойный цилиндр, состоящий из сердцевины (оптического световода) и оболочки, которые имеют разную оптическую плотность (показатель преломления): чем больше оптическая плотность материала, тем больше замедляется свет по сравнению со скоростью в вакууме. В сердцевине значение показателя преломления выше чем в оболочке. Передача сигнала по световоду осуществляется благодаря свойствам внутреннего отражения, которое обеспечивается как раз неравенством показателей преломления сердцевины и оболочки.

В качестве источника распространения света по оптическим кабелям выступает светодиод (или полупроводниковый лазер). Двухуровневое изменение интенсивности света (0–1) обеспечивает кодирование информации. Принимающий детектор на противоположном от источника конце кабеля преобразует световые сигналы в электрические.

Расстояние и скорость передачи данных ограничиваются как дисперсией, так и затуханием сигнала в волокне. Источники света генерируют спектр частот (длин волн), приводя к возникновению хроматической дисперсии из-за того, что волны света разной длины проходят через оптическое волокно с разной скоростью (нелинейность фазочастотной характеристики оптоволокна).

Составляющие луча света входят в оптическое волокно под разными углами, отражаясь от границы раздела сердцевины и оболочки также под разными углами. При этом возникает интерференция и частичное взаимное подавление составляющих луча. Оставшиеся составляющие части луча света формируют так называемые *моды*.

Различают *одномодовое* (SingleMode, SM) и *многомодовое* (MultiMode, MM) оптоволокно, которые отличаются принципом распространения светового сигнала внутри волокна.

Диаметр сердечника многомодового волокна почти на два порядка больше, чем длина световой волны (обычно 50 или 62,5 мкм), т.е. свет распространяется в волокне по нескольким независимым путям (модам), причём разные моды имеют разную длину. Из-за этого появляется межмодовая дисперсия передаваемого импульсного сигнала, снижая тем самым скорость передачи данных. Передаваемые импульсы накладываются друг на друга, приводя к уменьшению расстояния, на которое можно передать данные. Для снижения влияния наличия нескольких мод на величину дисперсии при большом диаметре сердцевины производителями предлагается специальное многомодовое оптоволокно с градиентным показателем преломления.

Меньший диаметр сердцевины (8–10 мкм) в одномодовом волокне позволяет распространяться только одной моде луча света по сердцевине вдоль оси оптоволокна. В таком кабеле скорость передачи выше, чем в многомодовым оптоволокне, больше расстояние, на которое могут быть переданы данные, отсутствует межмодовая дисперсия.

Расстояние передачи сигналов в локальных сетях по одномодовому волокну, определённое стандартом Gigabit Ethernet, составляет до 5 км, а в стандарте 10Gigabit Ethernet достигает 40 км. В линейных трактах телекоммуникационных систем на длине волны 1550 нм реализована передача данных на расстояние до 100 км без усиления и регенерации сигналов.

Типы одномодовых волокон:

- ступенчатое с несмещённой дисперсией (Step Index Single Mode Fiber, SMF или SM) — рекомендация ITU-T G.652, используется в большинстве систем оптической связи;
- со смещённой дисперсией (Dispersion Shifted Single Mode Fiber, DSF или DS) — рекомендация ITU-T G.653;
- с ненулевой смещённой дисперсией (Non-Zero Dispersion Shifted Single Mode Fiber, NZDSF, NZDS или NZ) — рекомендация ITU-T G.655.

Для приёма передаваемых по оптоволокну сигналов используют фотодиоды, которые работают на длинах волн 850, 1310 или 1550 нм, преобразуя принятые оптические импульсы в электрические.

Оптоволокно — традиционная физическая среда передачи данных по магистральным сетям. Способы его применения классифицируют по названию точки сопряжения с потребителем и объединяют названием типа FTTx — оптоволокно до точки «x», например:

- до жилого дома (Fiber To The Home, FTTH);
- до здания (Fiber To The Building, FTTB);
- до распределительного шкафа группы зданий (Fiber To The Curb, FTTC);

– до некоторого выносного модуля (Fiber To The Remote, FTTR);
– до сетевого узла (Fiber to the Node, FTTN).

Сравнение некоторых характеристик одномодовых и многомодовых технологий приведено в табл. 3.2.

Таблица 3.2

Сравнение одномодовых и многомодовых технологий

Параметры	Одномодовые	Многомодовые
Используемые длины волн	1,3 и 1,5 мкм	0,85 мкм, реже 1,3 мкм
Затухание, дБ/км	0,4–0,5	1,0–3,0
Тип передатчика	лазер, светодиод	светодиод
Толщина сердечника	8 мкм	50 или 62,5 мкм
Дальность передачи Fast Ethernet	около 20 км	до 2 км
Дальность передачи специально разработанных устройств Fast Ethernet	более 100 км	до 5 км
Возможная скорость передачи	10 Гбит и более	до 1 Гбит на ограниченной длине
Область применения	телекоммуникации	локальные сети

3.1.4. Диапазоны радиочастот и длин радиоволн

По регламенту международного союза электросвязи радиоволны разделены на диапазоны от 0.3×10^n Гц до 3×10^n Гц, где n — номер диапазона. Российский ГОСТ 24375-80 почти полностью повторяет эту классификацию. Сводная информация приведена в таблице 3.3.

3.1.5. Структурированная кабельная система

В основе функционирования любой телекоммуникационной сети лежит её кабельная система. Под кабельной системой понимается

Таблица 3.3

Диапазоны радиочастот

№	Обозначение ITU	Длины волн	Название волн	Частоты	Название частот
1	ELF	100 Мм–10 Мм	Декамегаметровые	3—30 Гц	Крайне низкие (КНЧ)
2	SLF	10 Мм–1 Мм	Мегаметровые	30—300 Гц	Сверхнизкие (СНЧ)
3	ULF	1000 км–100 км	Гектокилометровые	300—3000 Гц	Инфранизкие (ИНЧ)
4	VLF	100 км–10 км	Мириаметровые	3—30 кГц	Очень низкие (ОНЧ)
5	LF	10 км–1 км	Километровые	30—300 кГц	Низкие (НЧ)
6	MF	1000 м–100 м	Гектометровые	300—3000 кГц	Средние (СЧ)
7	HF	100 м–10 м	Декаметровые	3—30 МГц	Высокие (ВЧ)
8	VHF	10 м–1 м	Метровые волны	30—300 МГц	Очень высокие (ОВЧ)
9	UHF	1000 мм–100 мм	Дециметровые	300—3000 МГц	Ультравысокие (УВЧ)
10	SHF	100 мм–10 мм	Сантиметровые	3—30 ГГц	Сверхвысокие (СВЧ)
11	EHF	10 мм–1 мм	Миллиметровые	30—300 ГГц	Крайне высокие (КВЧ)
12	THF	1 мм–0,1 мм	Децимиллиметровые	300—3000 ГГц	Гипервысокие частоты, длинноволновая область инфракрасного излучения

система из кабелей и связанных с ним компонент. Компонентом кабельной системы является пассивное коммутационное оборудование, необходимое для соединения кабеля (телекоммуникационные розетки, кросс-панели и т.д.).

От качества кабельной системы сети напрямую зависят её надёжность и расширяемость в целом. Для обеспечения эффективности эксплуатации кабельных коммуникаций сети применяют принципы организации структурированной кабельной системы.

Структурированная кабельная система (СКС) — это иерархическая кабельная система здания или группы зданий, разделённая на стандартизованные структурные подсистемы. Фактически СКС может состоять из множества кабелей, разъёмов, кросс-панелей, розеток, модульных гнёзд, монтажных шкафов, коробов и пр. СКС включает в себя разные типы кабеля: сетевые, телефонные, кабели систем видеонаблюдения, сигнализации и др. Все перечисленные элементы интегрируются в единую систему и эксплуатируются согласно определённым правилам. Состав кабельной системы зависит от инфраструктуры используемых организацией информационных технологий, которая определяет содержание конкретного проекта построения СКС в соответствии с пожеланиями конечного пользователя и не зависит от применяемого в дальнейшем активного оборудования.

Важный принцип организации СКС — избыточность. Число элементов пассивного оборудования, устанавливаемого при прокладке кабельной системы, обычно существенно превышает необходимые в данный момент потребности заказчика. Такой подход позволяет в дальнейшем масштабировать нагрузку и произвести реконфигурацию сети.

СКС имеет чёткую иерархическую структуру. В соответствии международному стандарту ISO 11801 СКС имеет следующие основные подсистемы:

- *рабочего места* — набор соединительных кабелей и разъёмов оборудования пользователя, а также передающие устройства для его подключения к сети через информационные розетки;
- *горизонтальную* — совокупность коммутационного оборудования и кабельной системы этажа здания, ведущей к информационным розеткам рабочих мест пользователей;
- *вертикальную* (или внутренняя магистраль здания) — совокупность коммутационного оборудования и межэтажной кабельной системы здания;
- *административная* — физическое соединение линий подсистем, подключённых к коммутационным панелям здания;
- *подсистема оборудования* — активное сетевое (коммутационное) оборудование и элементы его подключения к коммутационным панелям здания;
- *внешнюю* (или внешняя магистраль здания) — совокупность комму-

тационного оборудования и внешних магистральных кабелей между кроссовой внешних магистралей и кроссовыми зданий (т.е. с помощью этой подсистемы здания, расположенные рядом, связываются в единую сеть);

В горизонтальной и вертикальной подсистемах СКС используется топология *звезда*. В горизонтальной подсистеме предпочтительно использовать кабели из неэкранированной витой пары (UTP) категории 5 с розетками, разъёмами и патч-панелями, удовлетворяющими той же категории. В вертикальных подсистемах рекомендуется использовать оптоволоконные кабели или экранированную витую пару (STP). Во внешней подсистеме между зданиями целесообразно использовать оптоволокно, так как оно обеспечивает наибольшую допустимую длину сегмента и пропускную способность.

Для соединения телекоммуникационного оборудования используется спецификация физического интерфейса *RJ*[1] *(Registered Jack)* (FCC, Part 68, Subpart F, Section 68.502[2]). Стандартные варианты этого разъёма называются RJ-11, RJ-14, RJ-25, RJ-45 и так далее. Разъёмы RJ (рис. 3.2) принадлежат к семейству модульных разъёмов, за исключением RJ-21.

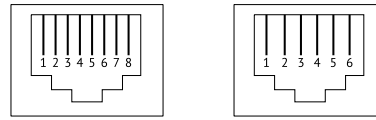


Рис. 3.1. Модульные розетки [1]

Термин *RJ-45* ошибочно употребляется для обозначения разъёма 8P8C, используемого в компьютерных сетях. На самом деле настоящий RJ-45 физически несовместим с 8P8C (8 контактов, 8 проводников), так как использует схему 8P2C (8 контактов, 2 проводника) с ключом. Ошибочное употребление термина RJ-45 вызвано, вероятно, тем, что настоящий RJ-45 не получил широкого применения, а также их внешним сходством.

[1] С этими стандартами связана большая путаница. Шестиместный разъём, часто применяемый в телефонии, может быть использован как RJ-11, RJ-14 или даже RJ-25, которые по сути являются названиями стандартов, использующих этот физический разъём. RJ-11 предполагает двужильное соединение, в то время как RJ-14 — четырёхжильное, а RJ-25 использует все шесть жил.

[2] FCC, Part 68, Subpart F, Section 68.502. URL: `http://www.fcc.gov/Bureaus/Engineering_Technology/Documents/cfr/1999/47cfr68.pdf`.

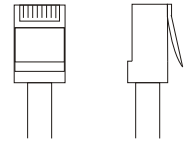


Рис. 3.2. Общий вид разъёма RJ-45 [1]

Для соединения оборудования в компьютерных и телефонных сетях чаще всего применяются восьмиконтактные модульные разъёмы RJ-45/8P8C (компьютерные), шестиконтактные RJ-12/6P6C (телефонные) и четырехконтактные RJ-11/6P4C (телефонные).

Четырёхконтактный модульный разъём RJ-11 используется в телефонии для соединения телефонных аппаратов с телефонными трубками. Шестиконтактный модульный разъём RJ-12 — в основном для соединения телефонных аппаратов с розеткой. Разъёмы RJ-11 и RJ-12 применяются с плоским 1–3-парным телефонным кабелем. При подключении шестиконтактного разъёма RJ-12 к аналоговому телефонному аппарату используются только два центральных контакта.

Использование контактов модульных соединителей, а также цветовая маркировка проводов стандартизованы. Каждая пара представляется двумя проводами, обозначаемыми Tip и Ring (условно — прямой и обратный провода), для которых определены цвет изоляции и номер контакта разъёма. Для обозначения пар кабеля используется цветовая маркировка (табл. 3.4).

Таблица 3.4

Цветовая маркировка витой пары

№ пары	Цвет: основной / полоски
1	синий / бело-синий
2	оранжевый / бело-оранжевый
3	зелёный / бело-зелёный
4	коричневый / бело-коричневый

Для разводки четырёхпарного кабеля UTP в разъёмах RJ-45 стандартом EIA/TIA-568 приняты две основные схемы распределения пар проводов по контактам: EIA/TIA-T568A и EIA/TIA-T568B (рис. 3.3, табл. 3.5).

Для соединения двух компьютеров (устройств) витой парой напрямую, без применения каких-либо дополнительных(промежуточных)

Таблица 3.5
Разводка контактов по схемам EIA/TIA-T568A и EIA/TIA-T568B

EIA/TIA-T568A			EIA/TIA-T568B		
№	Цвет	Пара	№	Цвет	Пара
1	бело-зелёный	3 (Tip)	1	бело-оранжевый	2 (Tip)
2	зелёный	3 (Ring)	2	оранжевый	2 (Ring)
3	бело-оранжевый	2 (Tip)	3	бело-зелёный	3 (Tip)
4	синий	1 (Ring)	4	синий	1 (Ring)
5	бело-синий	1 (Tip)	5	бело-синий	1 (Tip)
6	оранжевый	2 (Ring)	6	зелёный	3 (Ring)
7	бело-коричневый	4 (Tip)	7	бело-коричневый	4 (Tip)
8	коричневый	4 (Ring)	8	коричневый	4 (Ring)

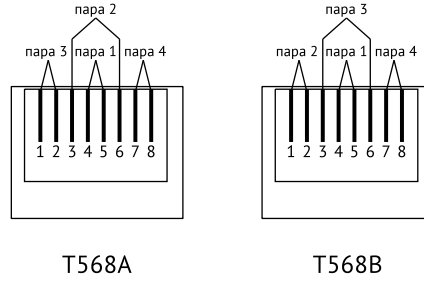


Рис. 3.3. Разводка контактов по схемам EIA/TIA-T568A и EIA/TIA-T568B [1]

устройств, служит «перекрёстный» (кроссовый) кабель. Концы такого кабеля обжимаются по разным стандартам (один EIA/TIA-568A, другой EIA/TIA-568B) (рис. 3.4).

Для подключения оптических кабелей применяются следующие разъёмы:

– FC коннектор (рис. 3.5а) — фиксируется накидной гайкой, используется для оконцовки одномодового оптоволоконного кабеля при подключении оптического оборудования к магистральным линиям;

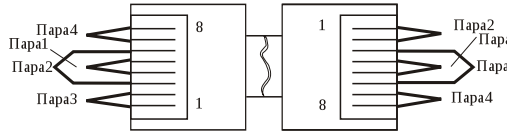
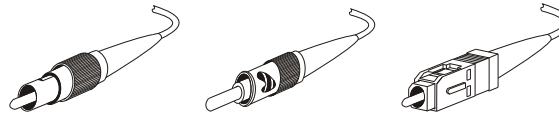


Рис. 3.4. Разводка кроссового кабеля [1]

– ST коннектор (рис. 3.5б) — используется для оконцовки многомодового оптоволоконного кабеля при подключении оптического оборудования к магистральным линиям, имеет гайку типа байонет для фиксации на розетке;
– SC коннектор (рис. 3.5в) — имеет прямоугольно-угловатую форму, фиксируются защёлкой, используется для оконцовки как одномодового, так и многомодового оптоволоконного кабеля внутри помещения;
– MT-RJ коннектор — одним разъёмом соединяются два волокна, в розетке фиксируется защёлкой рычажного типа, используется для внутренней кросс-коммутации;
– LC коннектор — миниатюрный разъём с диаметром керамического наконечника 1.25 мм и механизмом фиксации типа RJ-45, используется для коммутации многомодовых соединений внутри помещения.



(а) Оптический разъём FC **(б) Оптический разъём ST** **(в) Оптический разъём SC**

Рис. 3.5. Оптические разъёмы

Преимущества структурированной кабельной системы:
– длительный срок эксплуатации без модернизации (10–15 лет);
– надёжность;
– возможность наращивания мощности и лёгкость расширения сети без изменения существующей сети;
– универсальная среда передачи (единая кабельная система для передачи данных, голоса и видеосигнала).

Таким образом, СКС является современным сетевым решением, обеспечивающим здание надёжными и многофункциональными коммуникациями на довольно длительный срок.

3.2. Активное сетевое оборудование

Активное оборудование предназначено для выполнения всех необходимых действий, связанных с передачей данных. Активные устройства формируют, преобразуют, коммутируют, а также принимают сигнал с использованием внешнего (не передающегося в составе сигнала) источника энергии. Активные устройства можно, с некоторой долей условности, разделить на рабочие станции, повторители, концентраторы, коммутаторы, мосты и маршрутизаторы.

Повторитель (Repeator) представляет собой устройство для физического соединения двух или более сегментов кабеля локальной сети с целью увеличения общей длины сети.

В сетях на витой паре и оптоволокне повторитель является самым дешёвым вариантом связующего устройства и чаще называется концентратором.

Концентратор (Hub) представляет собой многопортовый повторитель с автосегментацией. Каждый порт имеет собственный трансивер — приёмник, передатчик и детектор коллизий. Получив сигнал от одной из подключённых к нему станций, концентратор транслирует его на все свои активные порты. Если на каком-либо из портов обнаружена неисправность, то этот порт автоматически отключается (сегментируется), а после её устранения снова становится активным.

Повторитель работает на уровне физических сигналов — закодированных битовых цепочек, анализ кадров не выполняется.

Сетевой коммутатор (Switch) — устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного сегмента. Коммутатор хранит в памяти таблицу MAC-адресов, в которой указывается соответствие MAC-адреса узла порту коммутатора. Коммутатор передаёт данные непосредственно получателю.

Мост (Bridge) делит разделяемую среду передачи сети на логические сегменты, передавая информацию из одного сегмента в другой только в том случае, если адрес узла назначения принадлежит другой подсети. Таким образом трафик одной подсети изолируется от трафика другой подсети, что увеличивает пропускную способность сети в целом и уменьшает возможность несанкционированного доступа в подсеть.

Маршрутизатор (Router) осуществляет связь разных типов сетей и обеспечивает доступ к глобальной сети, управляет трафиком на основе протокола сетевого уровня. Подобно повторителям, маршрутизаторы восстанавливают уровень и форму передаваемого сигнала.

Так же, как и мосты, они не передают адресату коллизии или повреждённые кадры, и из-за буферизации имеют задержку при передаче. Но в отличие от повторителей, мостов и коммутаторов, маршрутизаторы переформировывают передаваемые кадры Ethernet, а также могут поддерживать такие нетиповые функции, как подсчёт трафика, авторизация пользователей, ведение статистики и т.п.

Шлюз (Gateway) соединяет отдельные сегменты сети с разными типами системного и прикладного программного обеспечения.

Коммутаторы 3-го уровня (*маршрутизирующие коммутаторы* или *коммутирующие маршрутизаторы*) строятся на распределённой архитектуре — каждый порт имеет собственный специализированный процессор, отвечающий за анализ кадров и пакетов для определения их точки назначения, и общий управляющий процессор. Кадры, приходящие в порт и адресуемые (MAC-адресами) узлам той же подсети, но подключённым к другим портам, коммутируются (IP-заголовок не используется и не модифицируется). Кадры, приходящие на MAC-адрес порта, маршрутизируются — порт назначения определяется по IP-адресу назначения). Отличие от комбинации отдельного коммутатора с обычным маршрутизатором заключается в масштабировании пропускной способности каждой подсети: чем больше портов в неё входит, тем выше пропускная способность. Кроме того, и при коммутации может использоваться информация 3-го уровня (например, для фильтрации или приоритизации). Коммутаторы 3-го уровня в основном предназначены для организации связи подсетей в локальных сетях, и интерфейсов глобальных сетей они могут и не иметь.

Оптический медиаконвертер представляет собой устройство преобразования передаваемого сигнала из оптического (распространяемого по оптоволокну) в электрический (распространяемый по витой паре) и обратно. Медиаконвертеры используются в технологиях Fast Ethernet и Gigabit Ethernet, позволяют передавать данные со скоростью 10/100 и 1000 Мбит/с на расстояние от 2 до 120 км.

3.3. Модуляция сигналов

Модуляция — процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала (сообщения).

Несущая — высокочастотное колебание, выполняющее роль переносчика информации, заложенной в управляющем (модулирующем) сигнале.

В качестве несущего колебания наиболее часто используют *гармоническое колебание*. В зависимости от того, какой из параметров несущего колебания — амплитуда, частота или начальная фаза несущего колебания изменяется по закону передаваемого сообщения, различают виды

модуляции, соответственно, *амплитудная*, *частотная* или *фазовая*, а также их комбинации.

Сигнал, получаемый в процессе модуляции, называют *модулированным колебанием*, или радиосигналом.

Модуляция аналогового сигнала требуется в случае передачи низкочастотного (например, голосового) сигнала через канал высокочастотной области спектра — амплитуда высокочастотного несущего сигнала изменяется (модулируется) в соответствии с изменением низкочастотного сигнала.

Методы преобразования передаваемого сигнала делятся на временные и частотные. Эквивалентность таких частотно-временных изменений обеспечивается однозначно через преобразование Фурье.

Преобразование Фурье — операция, сопоставляющая функции вещественной переменной другую функцию вещественной переменной, которая описывает коэффициенты («амплитуды») при разложении исходной функции на элементарные составляющие — гармонические колебания с разными частотами.

Для функции $f(x) \in \mathbb{R}$ преобразование Фурье имеет вид:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{+\infty} f(x) e^{-ix\omega} dx.$$

Обратное преобразование Фурье функции $\hat{f}(\omega)$:

$$F^{-1}\left(\hat{f}(\omega)\right) = f(x) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega x} d\omega,$$

где $\hat{f}(\omega)$ — *спектральная плотность* или просто *спектр сигнала*.

Спектр некоторого периодического сигнала представляет собой дискретное множество гармонических колебаний, в сумме дающее исходный сигнал. Разложение некоторого сигнала на составляющие называется *спектральным*. График зависимости параметров сигнала от частоты называется *спектральной диаграммой*. *Спектр сигнала* — совокупность простых составляющих сигнала с определенными амплитудами, частотами и начальными фазами.

Изменение формы сигнала приводит к изменению его спектра (и наоборот).

Теорема Котельникова (теорема Найквиста–Шеннона или теорема отсчётов) гласит, что если аналоговый сигнал $x(t)$ имеет конечный (ограниченный по ширине) спектр, то он может быть однозначно восстановлен без потерь по своим отсчётам, взятым с частотой,

строго большей удвоенной верхней частоты f_c: $f > 2f_c$.

Для преобразования цифровых данных в аналоговый сигнал используются следующие основные технологии модуляции (или кодирования):
– амплитудная (Amplitude-Shift Keying, ASK),
– частотная (Frequency-Shift Keying, FSK),
– фазовая (Phase-Shift Keying, PSK).

3.3.1. Амплитудная модуляция

В этом типе модуляции представлению нуля или единицы соответствует наличие или отсутствие несущей частоты при постоянной амплитуде. Результирующий сигнал при этом имеет вид:

$$s(t) = \begin{cases} A\cos(2\pi f_c t), & \text{кодирует двоичную 1,} \\ 0, & \text{кодирует двоичный 0,} \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; t — время.

3.3.2. Частотная модуляция

Частотная модуляция бывает:
– бинарной (Binary FSK, BFSK),
– многочастотной (Multiple FSK, MFSK).

В первом случае два двоичных числа модулируются сигналами двух различных частот, расположенных около несущей. Результирующий сигнал при этом имеет вид:

$$s(t) = \begin{cases} A\cos(2\pi f_1 t), \\ A\cos(2\pi f_2 t), \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_1 и f_2 — частоты, смещённые от несущей частоты f_c на величины, равные по модулю, но противоположные по знаку; t — время.

Во втором случае за один раз пересылается более одного бита за счёт использования нескольких частот для модуляции сигнала. Сигнал при этом имеет вид:

$$s_i = A\cos(2\pi f_i t), \quad 1 \ll i \ll M,$$
$$f_i = f_c + (2i - 1 - M)f_d, \quad M = 2^L,$$

где $A\cos(2\pi f_i t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; f_d — разностная частота; M — число различных сигнальных

посылок; L — количество бит, переданных за один раз; t — время.

Бинарная частотная модуляция менее восприимчива к ошибкам, чем амплитудная модуляция. Многочастотная модуляция эффективнее бинарной, но и более подвержена ошибкам.

3.3.3. Фазовая модуляция

При фазовой модуляции для представления данных выполняется смещение несущего сигнала.

Фазовая модуляция бывает:

- двухуровневой (Binary PSK, BPSK),
- дифференциальной (Differential PSK, DPSK),
- квадратурной (Quadrature Phase-Shift Keying, QPSK),
- многоуровневой (Multiple FSK, MFSK).

В первом случае для представления двух двоичных цифр используются две фазы. Для одного периода передачи бита результирующий сигнал имеет вид:

$$s(t) = \begin{cases} A\cos(2\pi f_c t) \\ A\cos(2\pi f_c t + \pi) \end{cases} = \begin{cases} A\cos(2\pi f_c t), & \text{кодирует двоичную 1,} \\ -A\cos(2\pi f_c t), & \text{кодирует двоичный 0,} \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; t — время.

Во втором случае для представления двоичного нуля используется сигнал, фаза которого совпадает с фазой предыдущего сигнала, а для представления двоичной единицы — сигнал с фазой, противоположной фазе предыдущего. Такая схема называется *дифференциальной*, поскольку сдвиг фаз выполняется относительно предыдущего переданного бита, а не относительно какого-то эталонного сигнала.

В третьем случае за раз передаётся более одного бита, при этом вместо сдвига фазы на π, как в двухуровневой модуляции, используются сдвиги фаз, кратные $\pi/2$:

$$s(t) = \begin{cases} A\cos\left(2\pi f_c t + \dfrac{\pi}{4}\right), & \text{кодирует 11,} \\ A\cos\left(2\pi f_c t + \dfrac{3\pi}{4}\right), & \text{кодирует 10,} \\ A\cos\left(2\pi f_c t + \dfrac{5\pi}{4}\right), & \text{кодирует 00,} \\ A\cos\left(2\pi f_c t + \dfrac{7\pi}{4}\right), & \text{кодирует 01,} \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; t — время.

Схема работы многоуровневой фазовой модуляции схожа со а схемой работы квадратурной фазовой модуляции, но в каждый момент времени передаётся по три бита, используется восемь различных углов сдвига фаз, для каждого угла используется несколько амплитуд.

3.3.4. Квадратурная амплитудная модуляция

Схема работы квадратурной амплитудной модуляции (QAM) использует принципы функционирования амплитудной и фазовой модуляций. Два различных сигнала передаются одновременно на одной несущей частоте, но при этом задействованы две её амплитудно-модулированные копии, сдвинутые относительно друг друга на 90^o (т.е. находящиеся в квадратуре). Амплитуды копий несущей меняются дискретно, что приводит к образованию сигнала с дискретным изменением одновременно и амплитуды, и фазы. Приёмник полученные сигналы демодулирует и объединяет с целью восстановления исходного двоичного сигнала.

Исходя из таких соображений фазовую модуляцию можно рассматривать как частный случай квадратурной амплитудной модуляции.

В случае двухуровневой амплитудной модуляции (2QAM) каждый из двух потоков может находиться в одном из двух состояний, а объединённый поток — в одном из четырёх. В случае четырёхуровневой модуляции (т.е. четырёх различных уровней амплитуды, 4QAM) объединённый поток будет находиться уже в одном из 16 состояний. Чем больше число состояний, тем выше скорость передачи данных, возможная при определённой ширине полосы пропускания. Но чем больше число состояний, тем выше потенциальная частота возникновения ошибок из-за помех или поглощения.

3.3.5. Технология расширенного спектра

Основная идея метода состоит в том, чтобы распределить информационный сигнал по широкой полосе радиодиапазона, что в итоге позволит значительно усложнить подавление или перехват сигнала.

Расширение спектра скачкообразной перестройкой частоты (Frequency Hopping Spread Spectrum, FHSS)

Передача ведётся с постоянной сменой несущей в пределах широкого диапазона частот. В результате мощность сигнала распределяется по всему диапазону, а прослушивание какой-то определённой частоты даёт только небольшой шум. Последовательность несущих частот псевдослучайна и известна только передатчику и приёмнику. Попытка

подавления сигнала в каком-то узком диапазоне почти не ухудшает сигнал, так как подавляется только небольшая часть информации.

На каждой несущей частоте для передачи дискретной информации применяются стандартные методы модуляции — частотная или фазовая. Для синхронизации приёмника и передатчика в течение некоторого времени передаются синхронизирующие последовательности бит. Несущая частота меняется в соответствии с номерами частотных подканалов, вырабатываемых алгоритмом псевдослучайных чисел. Если частота смены подканалов ниже, чем скорость передачи данных в канале, то такой режим называют *медленным расширением спектра*, в противном случае — *быстрым расширением спектра*. Метод быстрого расширения спектра более устойчив к помехам, т.к. помехи, подавляющие сигнал в определённом подканале, не приводят к потере бита, поскольку его значение повторяется несколько раз в различных частотных подканалах. Метод медленного расширения спектра менее устойчив к помехам, но его проще реализовать.

Прямое последовательное расширение спектра (Direct Sequence Spread Spectrum, DSSS)

В методе прямого последовательного расширения спектра, в отличие от метода расширения спектра скачкообразной перестройкой частоты, весь частотный диапазон занимается не за счёт постоянных переключений с частоты на частоту, а за счёт того, что каждый бит информации заменяется последовательностью из N бит, что даёт увеличение тактовой скорости передачи сигналов в N раз и соответствующее расширение в N раз спектра сигнала.

Передача двоичной единицы заменяется передачей расширяющей последовательности. Двоичный нуль кодируется инверсным значением расширяющей последовательности. Количество бит в расширяющей последовательности определяет коэффициент расширения исходного кода. Для кодирования битов результирующего кода может использоваться любой вид модуляции. Чем больше коэффициент расширения, тем шире спектр результирующего сигнала и выше степень подавления помех. Но при этом растёт занимаемый каналом диапазон спектра.

Помехи искажают только определённые частоты спектра сигнала, поэтому приёмник с большой степенью вероятности может правильно распознавать передаваемую информацию.

Метод прямого последовательного расширения спектра в меньшей степени защищён от помех, чем метод быстрого расширения спектра, так как мощные помехи влияют на часть спектра, а значит, и на результат распознавания единиц или нулей.

3.4. Кодирование сигнала

Кодирование сигнала — это правило, описывающее отображение одного набора знаков в другой набор знаков. Тогда отображаемый набор знаков называется исходным алфавитом, а набор знаков, который используется для отображения, - кодовым алфавитом, или алфавитом для кодирования. При этом кодированию подлежат как отдельные символы исходного алфавита, так и их комбинации.

Рассмотрим наиболее распространённые методы кодирования.

3.4.1. Код NRZ и NRZI

Код *NRZ (Non Return to Zero)* — простейший двухуровневый код. Логической единице соответствует верхний уровень, логическому нулю — нижний, переходы электрического сигнала происходят на границе битов (рис. 3.6). Код NRZ отличается простотой и обеспечивает высокую скорость передачи, но не имеет синхронизации.

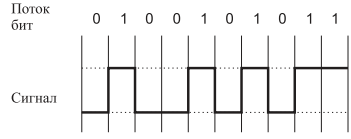


Рис. 3.6. Код NRZ

Код *NRZI (Non Return to Zero Invert to ones)* представляет собой модификацию кода NRZ. В этом двухуровневом коде принимается во внимание значение предыдущего бита. Уровень сигнала меняется, если текущий бит — единица, и повторяет предыдущий, если текущий бит имеет значение 0 (рис. 3.7). NRZI используется в основном для работы с оптоволоконной средой, в сетях 100Base-FX.

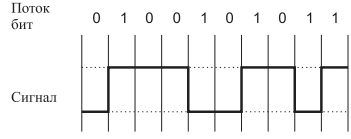


Рис. 3.7. Код NRZI

3.4.2. Код Rz

Код *RZ (Return to Zero)* обеспечивает возвращение к нулю после передачи каждого бита информации. RZ — трёхуровневый код. В центре бита всегда есть переход. Логической единице соответствует отрицательный импульс, логическому нулю — положительный (рис. 3.8). RZ — самосинхронизирующийся код, однако, он не даёт выигрыша в скорости. Код RZ нашёл применение в оптоволоконных сетях.

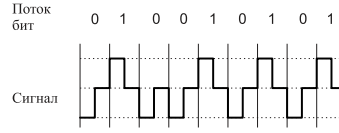


Рис. 3.8. Код Rz

3.4.3. Манчестерский код

Двухуровневый Манчестерский код широко используется в локальных сетях. Логической единице соответствует переход вниз в центре бита, логическому нулю — переход вверх (рис. 3.9). Манчестерский код является самосинхронизирующимся и обладает хорошей помехозащищённостью.

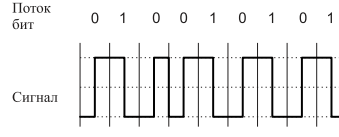


Рис. 3.9. Манчестерский код

3.4.4. Код MLT-3

Код *MLT-3 (Multi Level Transmission-3)* — трёхуровневый код. Как и в NRZI, логической единице соответствует смена уровня сигнала, а при передаче нуля сигнал не меняется (рис. 3.10). Изменение уровня сигнала происходит последовательно с учётом предыдущего перехода. Основной недостаток кода MLT-3 — отсутствие синхронизации. MLT-3 применяется в сетях 100Base-T на основе витой пары.

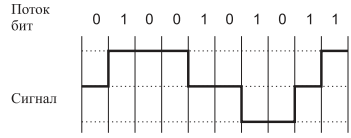


Рис. 3.10. Код MLT-3

3.5. Вопросы по разделу

1. Укажите функции и услуги физического уровня модели ISO/OSI.
2. Приведите классификацию среды передачи данных и ее основные характеристики.
3. Опишите схему организации СКС.
4. Опишите основные технологии модуляции (кодирования) сигнала.
5. Опишите принципы работы основных методов кодирования.

3.6. Примеры заданий

Задание 3.1 — Почему частота дискретизации аудиозаписи на CD составляет 44,1 кГц?

Ответ (Задание 3.1) — Считается, что верхняя граница восприятия звука у человека составляет около 20 кГц. По теореме Котельникова для передачи такого сигнала необходима частота отсчётов $2 \times 20 = 40$ кГц. Вероятно, за верхнюю границу восприятия звука было принято 22,05 кГц.

Задание 3.2 — Укажите ширину полосы пропускания для Ethernet 10Base-T.

Ответ (Задание 3.2) — При манчестерском кодировании каждому биту соответствует одно колебание, таким образом ширина полосы пропускания будет равняться 10 МГц.

3.7. Задания для самостоятельной работы

Задание 3.3 — Приведите временные диаграммы информационных сигналов с использованием различных кодов (NRZ, NRZI, AMI, Манчестерский код).

Задание 3.4 — Постройте разводку сети для здания факультета.

Глава 4. Канальный уровень

Глава посвящена подробному изучению канального уровня модели взаимодействия открытых систем. В ней рассмотрены основные модели и протоколы доступа к среде передачи данных. Внимание уделено также группе стандартов IEEE 802: технология доступа к среде Ethernet (и её производным), сети с маркерным доступом к среде (Token Bus, Token Ring, FDDI), технология 100VG-AnyLAN, технология доступа с виртуальными каналами (X.25, Frame Relay), технологии региональных сетей, технологии беспроводного доступа (Wi-Fi, WiMAX, Bluetooth).

В основу главы легли материалы из источников [1; 2; 13; 16—26; 34], а также материалы о технологии Bluetooth[1].

В результате освоения данной темы с учётом выполнения заданий раздела 9.2 студент должен:

знать:

- основные модели, технологии и протоколы доступа к среде передачи данных;
- структуру протоколов доступа к среде;

уметь:

- определять тип MAC-адреса;
- ориентироваться в стандартах IEEE 802;
- оценивать работоспособность сети, построенной на базе технологии Fast Ethernet;

владеть:

- способностью применять математический аппарат, в том числе с использованием вычислительной техники, для решения профессиональных задач, например, для оценки работоспособности сети, построенной на базе технологии Fast Ethernet.

[1] Bluetooth SIG, Inc. URL: `http://www.bluetooth.com/Bluetooth/`; Bluetooth в целом // Rainbow Technologies. 2005. URL: `http://www.rtcs.ru/article_detail.asp?id=331`; *Широков Ф.* Bluetooth: на пути к миру без проводов // Открытые системы. 2001. № 2. URL: `http://www.radioscanner.ru/info/article95/`; *Невдяев Л.* Bluetooth — королевская технология // Сети. 2000. № 10. URL: `http://www.osp.ru/nets/2000/10/141423/`; *Митилино С.* Беспроводные сети Bluetooth // Интернет и сети. 2002. URL: `http://itc.ua/node/11177/`; *Кессених В.*, *Иванов Е.*, *Кондрашов З.* Bluetooth: принципы построения и функционирования // Chip NEWS. 2001. URL: `http://www.chip-news.ru/archive/chipnews/200107/8.html`; *Редькин А.* Замена Bluetooth // Сети и Телекоммуникации. 2005. URL: `http://www.citforum.ncstu.ru/nets/wireless/wireless_usb/`.

4.1. Доступ к среде

При доступе к среде возникает проблема распределения одного широковещательного канала между несколькими пользователями. Можно выделить две схемы выделения канала: *статическую* и *динамическую*. Рассмотрим динамическое выделение канала.

4.1.1. Динамическое выделение канала

Рассмотрим вначале несколько моделей.

1. *Многостанционная модель*. В рамках этой модели рассматривается N независимых станций, каждая из которых порождает кадры с вероятностью $\lambda\Delta t$ за период Δt, где λ — некоторый параметр. После отправки кадра станция блокируется и ничего не предпринимает, пока кадр не будет успешно передан.
2. *Модель единого канала*. Для всех коммуникаций используется один канал. Все станции эквивалентны.
3. *Модель с коллизиями*. Если два кадра передаются одновременно, они перекрываются и возникает коллизия. Все станции могут детектировать коллизии. Эти кадры должны быть переданы ещё раз. За исключением коллизий, других ошибок нет.
4. *Временны́е модели*:
   (a) *Модель непрерывного времени*. Передача кадров может произойти в любой момент времени. Время непрерывно.
   (b) *Модель тактированного времени*. Время разбивается на дискретные интервалы — *такты (Slots)*. Передача кадров происходит всегда в начальный момент такта.
5. *Модели с несущей*:
   (a) *Модель с прослушиванием несущей*. Прежде чем использовать канал, станции запрашивают состояние канала. Если он занят, то все станции перестают его использовать до тех пор, пока он не освободится.
   (b) *Модель без прослушивания несущей*. Станции не запрашивают состояние канала перед началом передачи.

4.1.2. Протоколы множественного доступа

Рассмотрим некоторые модели протоколов с множественным доступом.

Семейство протоколов ALOHA

В 1970-х гг. в Гавайском университете под руководством Нормана Абрамсона была разработана система ALOHA. Она использовалась для наземной системы радиодоступа.

Центральный узел, называемый базовой станцией, принимает пакеты, передаваемые другими узлами на частоте $f_0 = 417$ МГц и ретранслирует эти пакеты на частоте $f_1 = 413$ МГц. Узлы сети ALOHA передавали пакеты со скоростью 9600 бит/с.

Узлы передают пакеты по общему каналу. Когда передача двух пакетов происходит одновременно, они искажают друг друга. Возникают коллизии. В начальной реализации сети ALOHA центральный узел подтверждает верно принятые пакеты. Когда узел не получает подтверждение за определённый промежуток времени, он полагает, что произошла коллизия, и передаёт пакет снова.

ALOHA не использует контроль несущей и не прекращает передачу пакета при обнаружении конфликта. Контроль несущей бесполезен, поскольку узлы расположены далеко друг от друга, и узел может завершить передачу прежде, чем другой узел заметит передачу. По тем же причинам обнаружение конфликтов слишком запаздывает.

Рассмотрим две версии протокола ALOHA: *чистую (Pure ALOHA)* и *тактированную (синхронную) (Slotted ALOHA)*. В первой используется модель непрерывного времени, а во второй — тактированного.

В модели *Чистая ALOHA* станция начинает передачу данных сразу же, как только у неё появляются данные. При возникновении коллизии посылающая станция ждёт случайный промежуток времени, а затем повторяет передачу этого кадра.

Таким образом, если станция начала передачу в то время, пока предыдущий кадр находится в канале, возникает коллизия. Оба пакета разрушаются и должны быть переданы повторно.

В модели *Тактированная ALOHA* время разбивается на дискретные интервалы. Передача может начаться только в начале такта. Когда у узла появляется новый пакет, он осуществляет его передачу в начале следующего такта. Если в течение этого временного интервала передаётся только один пакет, то передача является успешной. В противном случае возникает коллизия, и узел осуществляет повторную передачу через случайный период времени.

Для реализации тактированной версии протокола ALOHA необходимо приведение узлов к общему эталону времени для определения начала временны́х интервалов.

Протоколы множественного доступа с контролем несущей

Протоколы, в которых станции контролируют несущую, называются *протоколами с контролем несущей (Carrier Sense, CS)*.

Рассмотрим несколько видов протоколов семейства *CSMA (Carrier Sense Multiple Access)*.

1-устойчивый (1-persistent) CSMA. Когда станция готова к передаче данных, она прослушивает канал, чтобы определить, не передаёт ли данные кто-либо другой. Если канал занят, станция

ждёт, когда он освободится. Если же канал свободен, станция передаёт информацию. При возникновении коллизии станция ждёт случайный промежуток времени, а потом продолжает действовать по вышеописанному алгоритму. Протокол называется 1-устойчивый, потому что в случае свободного канала станция осуществляет передачу с вероятностью 1.

Неустойчивый (nonpersistent) CSMA. Этот случай немного отличается от предыдущего. Здесь опять перед передачей данных станция прослушивает канал. Но в случае, когда канал уже используется, станция ожидает случайный период времени и повторяет алгоритм.

p-устойчивый (p-persistent) CSMA. Данный вид применяется к тактированному каналу. Если канал свободен, то передача осуществляется с вероятностью p. Соответственно с вероятностью $q = 1 - p$ станция будет ждать следующего такта. Если канал свободен, то передача данных или ожидание следующего такта происходят с вероятностью p и q соответственно. Этот процесс продолжается до тех пор, пока либо кадр не будет передан, либо другая станция не начнёт передачу. В последнем случае возникает коллизия; станция ожидает случайный период времени и пытается снова осуществить передачу данных. Если же при прослушивании канала он оказывается занятым, то станция ждёт до начала следующего такта и повторяет алгоритм.

Устойчивый и неустойчивый CSMA являются непосредственным улучшением протоколов семейства ALOHA, поскольку в них для определения состояния канала осуществляется его прослушивание.

Протоколы множественного доступа с контролем несущей с определением коллизий (Carrier Sense Multiple Access with Collision Detection, CSMA/CD) являются дальнейшим улучшение протоколов, рассмотренных в предыдущем пункте. Здесь при возникновении коллизии станции сразу же прекращают передачу данных (вместо того, чтобы продолжать передачу, что бессмысленно). Это позволяет сэкономить и время, и полосу пропускания.

При обнаружении коллизии станция прекращает передачу данных и ждёт случайное время. По истечении данного времени станция опять пытается передать данные.

Разберём алгоритм определения коллизий подробнее. Пусть две станции начали передачу данных в один и тот же момент времени. Время определения коллизии будет зависеть от времени распространения сигнала между двумя этими станциями.

Обозначим через τ время прохождения сигнала между двумя наиболее удалёнными друг от друга станциями. Пусть первая станция начинает передачу данных в некоторый момент времени t_0. Если в промежуток времени $[t_0, t_0 + \tau)$ вторая станция тоже начнёт передачу, то она обнаружит коллизию. Чтобы коллизию обнаружила и первая

станция, сигнал должен вернуться обратно, то есть коллизия будет обнаружена в промежуток времени $[t_0, t_0 + \tau)$. Временной интервал 2τ называется *временем двойного оборота (Path Delay Value, PDV)*.

4.2. Группа стандартов IEEE 802

Как было сказано в разделе 2.2.3, стандарты IEEE 802 охватывают только два нижних уровня семиуровневой эталонной модели ISO/OSI — физический и канальный (рис. 4.1), отражая тем самым специфику локальных сетей.

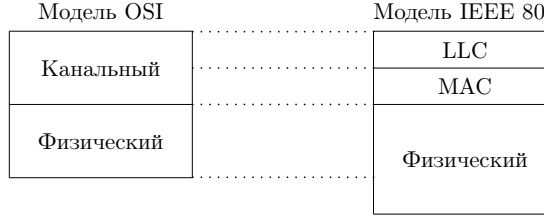


Рис. 4.1. Соответствие эталонных моделей ISO/OSI и IEEE 802

Многие сетевые стандарты IEEE легли в основу сетевых стандартов ISO и IEC. В 1980 г. в IEEE был организован комитет 802 по стандартизации локальных сетей. Результатом его деятельности стала разработка семейства протоколов IEEE 802, содержащих рекомендации по проектированию нижних уровней локальных сетей. Эти стандарты были созданы на основе распространённых фирменных стандартов сетей Arcnet, Ethernet, Token Ring.

4.2.1. Структура стандартов IEEE 802

Нумерация стандартов IEEE из серии 802 производится в соответствии со своей собственной схемой. Если за цифрой следует прописная буква, то это отдельный стандарт, если же за цифрой следует строчная буква, то это дополнение к стандарту или часть стандарта, обозначаемого несколькими цифрами.

Стандарты, разрабатываемые подкомитетом 802.1, носят общий для всех технологий характер. Именно в нём были разработаны общие определения локальных сетей и их свойств, обозначена связь эталонных моделей IEEE 802 и ISO/OSI. Также сюда входят стандарты *межсетевого (internetworking) взаимодействия*, описывающие взаимодействие разных технологий, и стандарты построения более сложных сетей. Это, например, стандарт IEEE 802.1D, описывающий

логику работы моста (коммутатора), стандарт IEEE 802.1Q, определяющий способ построения виртуальных локальных сетей (Virtual Local Area Network, VLAN) в сетях на основе коммутаторов.

Включение уровня LLC в стандарт IEEE позволило определить стандартный интерфейс на уровне MAC, однако используют настоящий LLC (т.е. LLC Type 2) только два протокола — SNA (Systems Network Architecture) и NetBEUI (NetBIOS Extended User Interface, расширенный пользовательский интерфейс NetBIOS), называемый также NetBIOS поверх LLC. Обычно применяются только заголовки LLC Type 1 в качестве заглушки (stub) для протоколов верхнего уровня. Стандартом LLC занимается подкомитет 802.2.

Стандарты 802.3, 802.4, 802.5 описывают технологии, созданные на основе фирменных технологий. Основу стандарта IEEE 802.3 составила технология Ethernet, разработанная компаниями DEC, Intel и Xerox. Стандарт IEEE 802.4 создан на основе технологии ArcNet фирмы Datapoint Corporation. Стандарт IEEE 802.5 базируется на технологии Token Ring компании IBM.

Исходные фирменные технологии и стандарты IEEE 802 в ряде случаев довольно долго существовали параллельно. Технология ArcNet так до конца и не была приведена в соответствие со стандартом IEEE 802.4[1]. Из-за того, что IBM регулярно вносит усовершенствования в технологию Token Ring, периодически возникают расхождения между стандартом IEEE 802.5 и данной технологией.

Сегодня комитет IEEE 802 включает следующие подкомитеты [16]:

802.1 Internetworking — объединение сетей.

802.1B Стандарт управления локальными/региональными сетями. Одобренный в 1992 г., он вместе с 802.1k лёг в основу ISO/IEC 15802-2.

802.1D Стандарт межсоединения локальных сетей с помощью мостов уровня MAC. Одобренный в 1990 г., он лёг в основу ISO/IEC 10038.

802.1E Стандарт на протоколы системной нагрузки для локальных и региональных сетей. Одобренный в 1990 г., он лёг в основу ISO/IEC 10038.

802.1F Стандарт определения управляющей информации для серии 802; одобрен в 1993 г.

802.1g Предложение по стандарту на удалённые мосты уровня MAC.

802.1H Рекомендуемые правила организации мостов MAC в сетях Ethernet 2.0; одобрены в 1995 г.

802.1i Стандарт на использование FDDI в качестве моста уровня MAC; одобрен в 1992 г. и включён в ISO/IEC 10038.

[1]Примерно в 1993 г. производство оборудования ArcNet прекращено.

802.1j Дополнение к 802.1D; одобрено в 1996 г. Данный стандарт описывает связь локальных сетей с помощью мостов уровня MAC.

802.1k Стандарт для локальных и региональных сетей на обнаружение и динамический контроль маршрутизации событий; одобрен в 1993 г. и вместе с 802.1B лёг в основу ISO/IEC 15802-2.

802.1m Описание соответствий для 802.1E, рассматривающее определения и правила управляемых объектов для протокола системной нагрузки; одобрено в 1993 году и включено в ISO/IEC 15802-4.

802.1p Предложение по стандарту для локальных и региональных сетей, касающееся ускорения обработки трафика и многоадресной фильтрации с помощью мостов уровня MAC.

802.1Q Предложение по стандарту на виртуальные локальные сети с мостами.

802.2 Logical Link Control, LLC — управление логической передачей данных. Стандарт для логического управления каналом связи локальных и региональных сетей, в основном с помощью мостов; лёг в основу ISO/IEC 8802-2. Текущая версия, одобренная в 1994 г., заменила более ранний стандарт 802.2 от 1989 г.

802.3 Стандарт на метод коллективного доступа для локальных сетей CSMA/CD и на физический уровень. Он положен в основу ISO/IEC 8802-3. Также его называют стандартом Ethernet.

802.3b Стандарт на устройства подключения к широкополосной среде передачи для 10Broad36. Одобрен в 1985 г. и включён в ISO/IEC 8802-3.

802.3c Стандарт на повторители в сети с немодулированной передачей на 10 Мбит/с. Одобрен в 1985 г. и включён в ISO/IEC 8802-3.

802.3d Стандарт на устройства подключения к среде передачи и спецификации среды с немодулированной передачей для каналов с оптическими повторителями. Одобрен в 1987 г. и включён в ISO/IEC 8802-3.

802.3e Стандарт на сигнализацию на физическом уровне, подключение к среде передачи и спецификации на среду с немодулированной передачей для сети на 1 Мбит/с, иными словами, 1Base5. Одобрен в 1987 году и включён в ISO/IEC 8802-3.

802.3h Стандарт на управление уровнем в сетях коллективного доступа CSMA/CD. Одобрен в 1990 г. и включён в ISO/IEC 8802-3.

802.3i Стандарт охватывает две области: многосегментную сеть немодулированной передачи на 10 Мбит/с и витую пару для сети 10BaseT. Одобрен в 1990 г. и включён в ISO/IEC 8802-3.

802.3j Стандарт на активные и пассивные сегменты в топологии звезда на 10 Мбит/с с использованием оптической среды пе-

редачи, т.е. 10BaseF. Одобрен в 1993 г. и включён в ISO/IEC 8802-3.

802.3k Стандарт на управление уровнем для повторителей в сети с немодулированной передачей на 10 Мбит/с. Одобрен в 1992 г. и включён в ISO/IEC 8802-3.

802.3l Описание соответствия для протоколов устройств подключения к среде передачи 10BaseT. Одобрено в 1992 г. и включено в ISO/IEC 8802-3.

802.3p Стандарт на управление уровнем для устройств подключения к среде с немодулированной передачей на 10 Мбит/с. Одобрен в 1993 г. и включён в ISO/IEC 8802-3.

802.3q Рекомендации по разработке управляемых объектов. Одобрены в 1993 г. и включены в ISO/IEC 8802-3.

802.3r Стандарт на метод коллективного доступа к среде передачи CSMA/CD, а также спецификации физического уровня для 10Base5. Пересмотрен в 1996 г.

802.3t Стандарт на поддержку 120-омных кабелей в сегментах с симплексными каналами 10BaseT. Включён в ISO/IEC 8802-3, одобрен в 1995 г.

802.3u Дополнение к 802.3, касающееся параметров MAC, физического уровня и повторителей на 100 Мбит/с, т. е. 100BaseT или, иначе, Fast Ethernet. Одобрено в 1995 г.

802.3v Стандарт для поддержки 150-омных кабелей в сегментах с каналами 10BaseT. Одобрен в 1995 г. и включён в ISO/IEC 8802-3.

802.3w Предложение по стандарту на усовершенствованные алгоритмы MAC.

802.3x Предложение по стандарту на полнодуплексный режим для 802.3.

802.3y Предложение по спецификации физического уровня для работы на 100 Мбит/с по двум парам категории 3 или ещё лучше сбалансированного кабеля на основе витой пары, т.е. 100BaseT2.

802.3z Предложение по стандарту на физический уровень, повторители и управляющие параметры для работы на 1000 Мбит/с, часто называемое Gigabit Ethernet.

802.4 Token Bus LAN. Стандарт на метод доступа к шине с передачей маркера и спецификации физического уровня. Одобрен в 1990 г.

802.5 Стандарт на методы доступа к кольцу с передачей маркера и спецификации физического уровня, т.е. на общую архитектуру Token Ring. Лёг в основу ISO/IEC 8802-5, текущая версия была одобрена в 1995 г.

802.6 Семейство стандартов на сеть с двойной шиной и распределённой очередью. Одобрено в 1990 г.

802.7 Broadband Technical Advisory Group — техническая консультационная группа по широкополосной передаче.

802.8 Fiber Optic Technical Advisory Group — техническая консультационная группа по волоконно-оптическим сетям.

802.9 Integrated Voice and Data Networks — интегрированные сети передачи данных и голоса. Стандарт на локальную сеть с интеграцией услуг (Integrated Services LAN) для подключения локальных сетей 802.x к общедоступным и частным магистральным сетям, таким как FDDI и ISDN. Одобрен в 1994 году и лёг в основу ISO/IEC 8802-9.

802.10 Стандарт на защиту локальных сетей Interoperable LAN Security, известный так же, как SILS. Одобрен в 1992 г.

802.11 Стандарт на беспроводные локальные сети (Wireless Local Area Networks, WLAN). Стандарт на уровень MAC и спецификации физического уровня для беспроводных локальных сетей. Предлагаемый проект рассчитан на диапазон 2,4 ГГц.

- **802.11a** Редакция стандарта 802.11 IEEE, в которой рассматриваются сети, работающие со скоростями до 54 Мбит/с по технологии DSSS (Direct Sequence Spread Spectrum).
- **802.11b** Редакция стандарта 802.11 IEEE, в которой рассматриваются сети, работающие со скоростями до 11 Мбит/с по технологии DSSS (Direct Sequence Spread Spectrum).
- **802.11e** Редакция стандарта 802.11 IEEE по качеству услуг (Quality of Service, QoS).
- **802.11f** Редакция стандарта 802.11 IEEE, определяющая протокол взаимодействия точек доступа (Inter-Access Point Protocol).
- **802.11g** Редакция стандарта 802.11 IEEE, в которой рассматриваются сети, работающие со скоростями до 54 Мбит/с по технологии DSSS, обратно совместимые со стандартом 802.11b.
- **802.11h** Редакция стандарта 802.11 IEEE, определяющая управляемый спектр для 802.11a (Managed Spectrum for 802.11a).
- **802.11i** Стандарт IEEE, относящийся к безопасности беспроводных сетей. В нём объединены протоколы 802.1x и TKIP/CCMP, что позволяет обеспечить аутентификацию пользователей, конфиденциальность и целостность данных в беспроводных локальных сетях.

802.12 Demand Priority Access LAN, 100VG-AnyLAN — локальные сети с методом доступа по требованию с приоритетами.

802.15 Стандарт беспроводных персональных сетей (Wireless Personal Area Networks, WPAN), работающих на ограниченных расстояниях.

- **802.15.1** Bluetooth (базируется на спецификациях Bluetooth v1.x).
- **802.15.3** Стандарт беспроводных сетей, являющийся прямым наследником Bluetooth (частота 2,4 ГГц). Определяет беспроводные персональные сети со скоростью передачи данных 55 Мбит/с для мультимедийных приложений.
- **802.15.4 (ZigBee)** Стандарт определяет спецификации физического слоя и протокол управления доступом (MAC), предлагая

поддержку различных топологий сетей.

802.15.4a Технология сверхширокополосной связи (Ultra Wideband, UWB), при помощи которой можно создавать специальные сети, где несколько сверхширокополосных устройств смогут поддерживать связь между любыми узлами.

802.16 Стандарт широкополосной беспроводной связи (Broadband Wireless Access, BWA), в частности, Worldwide Interoperability for Microwave Access — WiMax.

802.17 Стандарт на адаптивные, кольцевые, высокоскоростные сети.

802.18 Техническая консультативная группа по регулированию радиотехнологий (Radio Regulatory Technical Advisory Group, RRTAG).

802.19 Техническая консультативная группа по совместимости (Coexistence Technical Advisory Group, CoTAG).

802.20 Стандарт на мобильный широкополосный беспроводной доступ (Mobile Broadband Wireless Access, MBWA).

802.21 Стандарт на услуги эстафетной передачи соединения независимо от среды (Media Independent Handover Services, MIHS).

802.22 Стандарт на беспроводные региональные сети (Wireless Regional Area Networks, WRAN).

4.2.2. Протокол MAC

Адресация MAC-уровня

Протоколы семейства IEEE 802 используют 48-битную схему адресации MAC-уровня (рис. 4.2). IEEE также предлагал 16-разрядный MAC-адрес, но он не получил большого распространения.

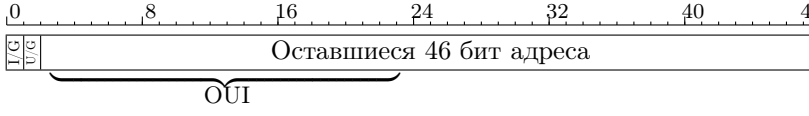


Рис. 4.2. Структура MAC-адреса IEEE. I/G: $= 0$ — **индивидуальный адрес,** $= 1$ — **групповой адрес; U/G:** $= 0$ — **глобально администрируемый адрес,** $= 1$ — **локально администрируемый адрес**

Первый бит MAC-адреса получателя называется *индивидуальным/групповым битом (Individual/Group, I/G)*. Если он установлен в 0, то кадр послан определённой рабочей станции, если же он установлен в 1, то кадр является широковещательным (поэтому данный бит также называют *широковещательным битом*). Если и все остальные биты адреса установлены в 1, то широковещательный кадр предназначен всем станциям, в противном случае мы имеем дело с *групповой (multicast)*

рассылкой кадра на выделенное подмножество станций (станции должны быть сконфигурированы для приёма групповых адресов).

В адресе источника первый бит называется *индикатором маршрута от источника (Source Route Indicator)*.

Три старших байта адреса называют *зашитым адресом (Burned in Address, BIA)* или *уникальным идентификатором организации (Organizationally Unique Identifier, OUI)*. Этот идентификатор выдаётся каждому производителю оборудования (распределением OUI занимался сначала Xerox, теперь эти полномочия делегированы IEEE). За уникальность младших трёх байт адреса отвечает сам производитель.

Второй бит адреса определяет способ назначения адреса. Если он выставлен в 0, то адрес является *централизованно или глобально администрируемым (Universally/Globally Administered)*. В этом случае сохраняется адрес, заданный производителем. Если же этот бит установлен в 1, то адрес является *локально администрируемым (Locally Administered Address, LAA)*, т.е. текущий адрес заменяет адрес, установленный производителем.

IBM ввела локально администрируемые адреса, чтобы пользователи могли работать с адресом сети SNA при обращении извне к этой сети. Очевидный недостаток этих адресов — возможность появления в сети дублированных адресов. Локально администрируемые адреса допустимы и в Ethernet.

Следует отметить специфику MAC-адреса для Ethernet (рис. 4.3). В стандарте Ethernet младший бит байта отображается в самой левой позиции, а старший бит — в самой правой. При этом порядок следования байтов остаётся традиционным. Обычно же младшим считается самый правый бит байта, а старшим — самый левый.

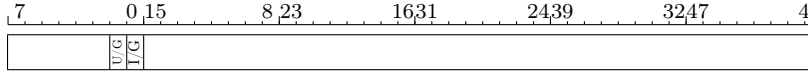


Рис. 4.3. Структура MAC-адреса Ethernet

4.2.3. Протокол IEEE 802.2 LLC

Уровень управления логическим каналом (Logical Link Control, LLC) отвечает за передачу кадров между узлами с различной степенью надёжности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем.

Протоколы сетевого уровня передают данные для протокола LLC: свой пакет, адресную информацию об узле назначения, требования к качеству транспортных услуг, которое должен обеспечить протокол LLC. Протокол LLC помещает пакет протокола верхнего уровня в свой

кадр, дополняя необходимыми служебными полями. Потом протокол LLC передаёт свой кадр соответствующему протоколу уровня MAC.

В основу протокола LLC положен протокол *HDLC (Hight-level Data Link Control Procedure)*. Поскольку данный протокол пришлось сопрягать с разными фирменными протоколами, то на уровне LLC пришлось ввести три типа процедур, к одной из которых может обращаться протокол сетевого уровня.

В соответствии со стандартом IEEE 802.2 уровень LLC предоставляет верхним уровням три типа процедур:

- *LLC1, Type 1, Conectionless* — процедура без установления соединения и без подтверждения;
- *LLC2, Type 2, Connection-Oriented* — процедура с установлением соединения и с подтверждением;
- *LLC3, Type 3* — процедура без установления соединения, но с подтверждением.

Процедура без установления соединения и без подтверждения LLC1 представляет собой заглушку (stub) для мультиплексирования или идентифицирует протокол следующего уровня. Она позволяет передавать данные с минимумом издержек. Это датаграммный режим работы. В этом режиме работают такие стеки протоколов, как TCP/IP, IPX/SPX.

Процедура с установлением соединения и с подтверждением LLC2 предоставляет функции транспортного уровня на уровне DLC без участия промежуточного сетевого уровня. Она даёт возможность установить логическое соединение перед началом передачи блока данных и выполнить процедуры восстановления после ошибок и упорядочивания потока этих блоков в рамках установленного соединения.

Протокол LLC Type 2 применяется на сегодняшний день только в двух случаях:

- в стеке протоколов SNA, когда на нижнем уровне используется Token Ring;
- в NetBEUI (NetBIOS поверх LLC).

Процедура без установления соединения, но с подтверждением LLC3 используется в случае, когда временны́е издержки установления логического соединения перед отправкой данных не приемлемы, а подтверждение приёма данных необходимо (например, при использовании сетей в системах реального времени).

По своему назначению все кадры уровня LLC подразделяются на три типа.

- *Информационные кадры (Information, I-frame)* предназначены для передачи информации в процедурах с установлением логического соединения LLC2 и должны обязательно содержать поле информации. В процессе передачи информационных блоков осуществляется их нумерация в режиме скользящего окна.

– *Управляющие кадры (Supervisory, S-frame)* предназначены для передачи команд и ответов в процедурах с установлением логического соединения LLC2, в том числе и запросов на повторную передачу искажённых информационных блоков.
– *Ненумерованные кадры (Unnumbered, U-frame)* предназначены для передачи ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, а в процедурах с установлением логического соединения LLC2 — установление и разъединение логического соединения, а также информирование об ошибках.

Все типы кадров LLC имеют единый формат (рис. 4.4). Кадр обрамляется двумя однобайтовыми полями *Флаг*, содержащими значение 0b01111110. Флаги используются для определения границ кадра LLC. При вложении кадра LLC в кадр MAC флаги отбрасываются.

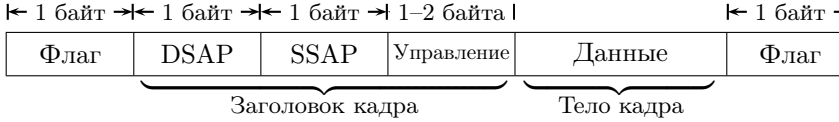


Рис. 4.4. Формат кадра LLC

Заголовок кадра LLC состоит из трёх полей:
– точки доступа к службе получателя (Destination Service Access Point, DSAP);
– точки доступа к службе источника (Source Service Access Point, SSAP);
– управляющего поля (Control).

В *поле данных* вкладываются пакеты протоколов вышележащих уровней. Поле данных может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.

Поля DSAP и SSAP имеют размер 1 байт каждое (рис. 4.5). Они служат для идентификации протокола верхнего уровня, инкапсулировавшего данные в кадр LLC. Служба может иметь несколько SAP, что может быть использовано протоколом узла отправителя в специальных целях, например, для уведомления узла-получателя о переходе протокола-отправителя в некий специфический режим работы.

Поле управления длиной 1 или 2 байта имеет разную структуру в зависимости от типа кадра LLC (рис. 4.6).

В LLC1 используется только один тип кадра — *ненумерованный* (рис. 4.6в). У этого кадра поле управления имеет длину 1 байт. Все подполя поля управления ненумерованных кадров в этом режиме имеют нулевые значения, так что значимыми остаются только два первых бита, используемые как признак типа кадра.

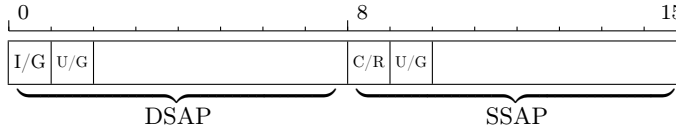
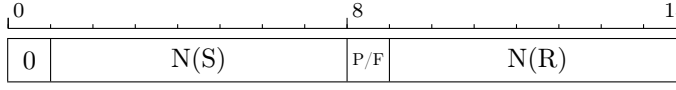
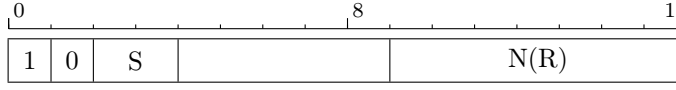


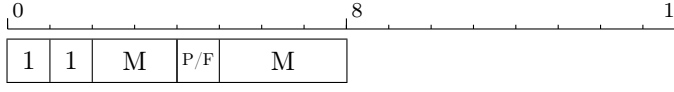
Рис. 4.5. Структура полей SAP. U/G: = 0 — глобально администрируемая точка доступа к службе, = 1 — локально администрируемая точка доступа к службе; I/G: = 0 — индивидуальная точка доступа к службе, = 1 — групповая точка доступа к службе; C/R: = 0 — команда, = 1 — отклик



(а) Информационный кадр



(б) Управляющий кадр



(в) Ненумерованный кадр

Рис. 4.6. Структура поля управления кадров LLC: P/F — бит опрос/завершение; N(S) — порядковый номер отправки; N(R) — порядковый номер получения

В LLC2 используются все три типа кадров. В этом режиме кадры делятся на команды и ответы. Бит *P/F (Poll/Final — опрос/завершение)* имеет следующее значение: в командах он называется битом *Poll* и требует ответа на команду, а в ответах он называется битом *Final* и говорит о том, что ответ состоит из одного кадра.

Ненумерованные кадры используются на начальной стадии взаимодействия двух узлов — стадии установления соединения по протоколу LLC2. Поле *M* ненумерованных кадров определяет несколько типов команд, которыми пользуются два узла на этапе установления соединением, например:

- *установка сбалансированного асинхронного расширенного режима (Set Asynchronouse Balanced Mode Extended, SABME)*, что является запросом на установление соединением, причём расширенный режим обозначает использование двухбайтных полей управления для кадров остальных двух типов;
- *ненумерованное подтверждение (UA)* служит для подтверждения установления или разрыва соединения;
- *сброс соединения (REST)* является запросом на разрыв соединения.

После установления соединения данные начинают передаваться в *информационных кадрах* (рис. 4.6а). Логический канал протокола LLC2 является дуплексным. В дуплексном режиме положительные квитанции на кадры также передаются в *информационных кадрах*.

Управляющие кадры (рис. 4.6б) используются для передачи отрицательных квитанций или в полудуплексном режиме (когда нет потока кадров в обратном направлении).

В состав управляющих кадров входят следующие:

- отказ (Reject, REJ);
- приёмник не готов (Receiver Not Ready, RNR);
- приёмник готов (Receiver Ready, RR).

Команда *RR* часто используется как положительная квитанция в случае, если поток данных от приёмника к передатчику отсутствует, а команда *RNR* — для замедления потока данных к приёмнику. Это может быть необходимо, когда приёмник не успевает обработать поток кадров. Получение кадра *RNR* требует от отправителя полной остановки передачи до получения кадра *RR*. С помощью этих команд осуществляется управление потоком данных.

Поле *N(S)* указывает номер отправленного кадра, а поле *N(R)* — номер кадра, который приёмник ожидает получить от передатчика следующим. Поскольку длина каждого из этих полей равна 7 бит, то они могут принимать значения в диапазоне от 0 до 127.

Подтверждение отсылается получателем отправителю с порядковым номером, равным номеру следующего кадра, который ожидает принять получатель от отправителя. Если же приёмник получает от отправителя кадр с номером, не равным ожидаемому, то этот кадр отбрасывается и посылается отрицательная квитанция *REJ* с номером ожидаемого кадра. При получении отрицательной квитанции передатчик обязан повторить передачу кадра с требуемым номером, а также всех кадров с большими номерами, которые он уже успел отослать.

При работе протокола LLC2 используется скользящее окно размером в 127 кадров.

4.3. Технология Ethernet

4.3.1. Метод доступа CSMA/CD

Метод доступа к среде в технологии Ethernet является вариантом метода CSMA/CD, а именно метод CSMA/CD *с двоичной экспоненциальной отсрочкой (Binary Exponential Backoff)*.

Если станция готова к передаче данных, то она действует по следующему алгоритму:

1. Станция ожидает освобождение канала.
2. После освобождения канала перед непосредственной передачей станция выдерживает паузу, называемую *межкадровым интервалом (Inter Packet Gap, IPG)*, длительность которой равна времени передачи 96 бит. Для скорости 10 Мбит/с пауза составляет 9,6 мкс, а для скорости 100 Мбит/с — 0,96 мкс. Эта пауза нужна для предотвращения монопольного захвата сети одной станцией.
   Во время передачи станция продолжает контролировать состояние канала. Если передаваемый и наблюдаемый сигналы отличаются, то считается, что обнаружена коллизия.
3. Если конфликт выявляется во время передачи преамбулы (подробнее структура кадра Ethernet будет рассмотрена в разделе 4.3.2), то оставшаяся часть преамбулы всё равно передаётся, чтобы усилить сигнал коллизии. Если конфликт возникает во время пересылки остальной части кадра, станция пересылает последовательность из 32 бит, называемую *jam-последовательностью*.
4. После прекращения передачи пакета станция ожидает случайное время, затем переходит к шагу 1.

Рассмотрим *алгоритм выбора случайного времени ожидания*. После возникновения коллизии время разбивается на дискретные промежутки, длительность каждого из которых устанавливается равной 512 bt[1]. Назовём этот промежуток *интервалом отсрочки*.

После первой коллизии станции ожидают 0 или 1 интервал отсрочки. После второй период ожидания длится 0, 1, 2 или 3 интервала отсрочки. Иными словами, выбирается количество интервалов отсрочки из интервала $[0, 2^n - 1]$, где n — номер попытки. После десятой попытки верхняя граница интервала фиксируется. После шестнадцатой попытки передатчик должен прекратить передачу и отбросить этот кадр.

[1]Битовый интервал (bit time, bt) — время между появлением двух последовательных бит данных на кабеле. Например, для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс.

4.3.2. Форматы кадров Ethernet

В процессе развития Ethernet и стандарта IEEE 802.3 было предложено 4 варианта формата кадра. В 1980 г. консорциум трёх фирм DEC, Intel, Xerox представил на рассмотрение комитета 802.3 свою версию стандарта Ethernet (тип кадра Ethernet DIX), но комитет принял стандарт, отличающийся деталями (в том числе и форматом кадра) от предложения DIX (тип кадра 802.3/LLC). Novell, являющаяся в то время лидером сетевой индустрии в области персональных компьютеров, предложила свой формат кадра (Raw 802.3). Четвёртый вариант был предложен комитетом 802.2 для ликвидации недостатков формата кадра 802.3/LLC и приведения всех форматов кадров к общему виду (тип кадра Ethernet SNAP).

Рис. 4.7. Формат кадра Ethernet

Каждый кадр начинается с *преамбулы (Preamble)* (длина 7 байт), заполненной шаблоном 0b10101010 (для синхронизации источника и получателя). После преамбулы идёт байт *начального ограничителя кадра (Start of Frame Delimiter, SFD)*, содержащий последовательность 0b10101011 и указывающий на начало собственно кадра.

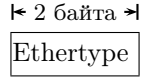
Далее идут поля *адрес получателя (Destination Address, DA)* и *адрес отправителя (Source Address, SA)*. В Ethernet используют 48-битные адреса MAC-уровня IEEE (см. раздел 4.2.2).

Следующее поле имеет разный смысл и разную длину в зависимости от типа кадра:

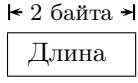
– Тип кадра Ethernet DIX.
  Тип кадра Ethernet DIX — изначальный тип кадра стандарта Ethernet. Этот тип кадра носит также названия Ethertype, Ethernet II (в терминологии NetWare). После поля адреса источника этот тип кадра содержит 16-битное поле *типа (Ethertype)*, идентифицирующее инкапсулированный в кадре протокол верхнего уровня (рис. 4.8а).
– Тип кадра Raw 802.3.
  Этот тип кадра предложен компанией Novell для своей системы NetWare. Он также носит названия Novell 802.3, Ethernet 802.3 (в терминологии NetWare).
  За адресом источника он содержит 16-битное поле *длины (Length,*

L), определяющее число байт, следующее за полем длины (без учёта поля контрольной суммы) (рис. 4.8б).
В этот тип кадра всегда вкладывается пакет протокола IPX. Первые два байта заголовка протокола IPX содержат контрольную сумму датаграммы IPX. Однако по умолчанию это поле не используется и выставлено в 0xFFFF.

– Тип кадра 802.3/LLC.
Поскольку группа стандартов IEEE 802 разделяет канальный уровень на подуровни MAC и LLC, то в кадр MAC-подуровня вкладывается кадр LLC-подуровня (см. раздел 4.2.3).

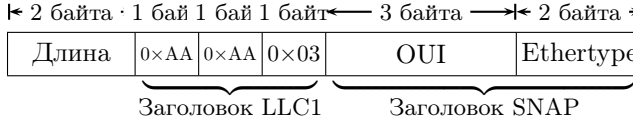


(а) Тип кадра Ethernet DIX



(б) Тип кадра Raw 802.3

(в) Тип кадра 802.3/LLC



(г) Тип кадра Ethernet SNAP

Рис. 4.8. Типы кадров Ethernet (только поле, зависящее от типа кадра)

За полем адреса источника идёт 16-битное поле *длины (Length, L)*, определяющее число байт, следующее за полем длины (без учёта поля контрольной суммы).

За ним следует заголовок LLC (рис. 4.8в). Он состоит из 8-разрядных полей *точки доступа к услуге источника (Source Service Access Point, SSAP)* и *точки доступа к услуге получателя (Destination Service Access Point, DSAP)*, а также поля управления, имеющего длину 8 или 16 бит, в зависимости от типа протокола LLC.

Поля SSAP и DSAP размером по 6 бит (см. рис. 4.5) предназначены для описания типа протокола следующего уровня. Но при такой разрядности можно указать не более 64 различных протоколов. Таким образом, недостаточный размер полей SAP создаёт трудности при применении этого типа кадра (например, нет типа SAP для протокола ARP).

– Тип кадра Ethernet SNAP.

Преждевременная стандартизация протокола LLC привела к значительным трудностям в применении типа кадра 802.3/LLC. Для решения этой проблемы комитетом 802.2 был предложен тип кадра *Ethernet SNAP (SubNetwork Access Protocol, протокол доступа к подсети)*.

Кадр Ethernet SNAP является расширением кадра 802.3/LLC за счёт введения дополнительного заголовка протокола SNAP. Заголовок SNAP состоит из 3-байтного поля *уникального идентификатора организации (Organizationally Unique Identifier, OUI)* и 2-байтного поля *типа (Type, Ethertype)*. Тип идентифицирует протокол верхнего уровня, а поле OUI определяет идентификатор организации, контролирующей назначение кодов типа протокола. Коды протоколов для стандартов IEEE 802 контролирует IEEE, имеющая код OUI, равный 0x000000. Для этого кода OUI поле типа для Ethernet SNAP совпадает со значением типа для Ethernet DIX.

Протокол SNAP вкладывается в протокол LLC1. Код SAP для него — 0xAA. Поле управления устанавливается в 0x03, что соответствует использованию ненумерованных кадров (см. рис. 4.6в, учитывая обратный порядок записи битов в байте в протоколе Ethernet).

Далее идёт поле *данных (Data)*. Если длина поля данных недостаточна для получения минимальной длины кадра, то вводится дополнительное поле *заполнения (Padding)*, призванное обеспечить минимальную длину кадра.

В конце кадра идёт 32-битное поле *контрольной суммы (Frame Check Sequence, FCS)*. Контрольная сумма вычисляется по алгоритму CRC-32.

Размер кадра Ethernet от 64 до 1518 байт (без учёта преамбулы, но с учётом поля контрольной суммы) (см. рис. 4.7).

Алгоритм автоматического распознавания разных типов кадров Ethernet достаточно прост.

Поле, следующее за полем адреса источника, имеет длину 2 байта и может быть либо полем Ethertype, либо полем длины данных. Максимальная длина поля данных равна 1500 байт (0x05DC). Значение поля Ethertype всегда больше, чем 0x05DC. Следовательно, если значение поля больше, чем 0x05DC, то мы имеем кадр Ethernet DIX. В противном случае — поле длины.

Если следующие за полем длины два байта выставлены в 0xFFFF, то это кадр Raw 802.3. В противном случае мы имеем либо кадр типа 802.3/LLC, либо кадр типа Ethernet SNAP, которые можно различить по значению полей SSAP и DSAP. Если они выставлены в 0xAA, то имеем кадр Ethernet SNAP, иначе — кадр типа 802.3/LLC.

В таблице 4.1 приведены данные об использовании разных кадров Ethernet протоколами более высоких уровней.

Таблица 4.1

Использование разных типов кадров Ethernet протоколами высших уровней

Тип кадра	Протоколы
Ethernet DIX	IPX, IP, AppleTalk Phase I
Raw 802.3	IPX
802.3/LLC	IPX, NetBEUI
Ethernet SNAP	IPX, IP, AppleTalk Phase II

4.3.3. Технология Fast Ethernet

В 1992 г. группой производителей сетевого оборудования было образовано некоммерческое объединение Fast Ethernet Alliance, целью которого стала разработка стандарта на технологию, обобщающую достижения отдельных компаний в области Ethernet-преемственного высокоскоростного стандарта. Технология получила название Fast Ethernet, и в 1995 г. комитет IEEE принял её спецификацию в качестве стандарта IEEE 802.3u.

Технология Fast Ethernet представляет собой эволюционное развитие технологии Ethernet. В данной технологии такие же формат кадра, механизм доступа к среде CSMA/CD и топология, как и в Ethernet. Эволюция коснулась нескольких элементов конфигурации средств физического уровня, включая типы применяемого кабеля, длину сегментов и количество концентраторов, что позволило увеличить пропускную способность.

Технология Fast Ethernet может использовать различные типы кабеля: витую пару разной категории, оптоволокно, причём по сравнению с Ethernet меняется как количество используемых проводников (для витой пары), так и методы кодирования. При кодировании сигнала применяются методы NRZI и MLT-3, при физическом кодировании — 4B/5B и 8B/6T.

В технологии Fast Ethernet реализована возможность выбора наиболее эффективного режима работы двух взаимодействующих портов: скорость передачи — 10 или 100 Мбит/с, вид передачи данных — дуплекс (full-duplex mode) или полудуплекс. Кроме того, во время выбора режима работы осуществляется проверка целостности линии. В режиме full-duplex вместо CSMA/CD используется соединение P2P (точка–точка) и отсутствует понятие коллизий — каждый узел может одновременно передавать и принимать кадры данных. Работа в данном режиме возможна только при соединении сетевого адаптера с коммутатором или же при непосредственном соединении коммутаторов.

Физическое соединение

Физически и логически сети на базе технологии Fast Ethernet имеют топологию звезда.

В качестве физической среды может использоваться:

- витая пара:
  - 2 пары UTP CAT.5 (100Base-TX),
  - 2 пары STP (100Base-TX),
  - 4 пары UTP CAT. 3, 4, 5 (100Base-T4);
- оптоволокно (100Base-FX).

Достоинства и недостатки технологии Fast Ethernet

Достоинства:

- увеличение пропускной способности сегментов сети до 100 Мбит/с;
- сохранение совместимости с методом случайного доступа CSMA/CD;
- сохранение формата кадра Ethernet;
- сохранение топологии звезда при построении сети;
- поддержка традиционных сред передачи данных — витой пары и оптоволоконного кабеля.

Недостатки (унаследованы от Ethernet):

- большие задержки доступа к среде при коэффициенте использования среды выше 30–40%, что связано с применением алгоритма доступа CSMA/CD;
- небольшие расстояния между узлами даже при использования оптоволокна, что связано с работой метода обнаружения коллизий;
- отсутствие механизмов выбора резервных связей;

– отсутствие поддержки приоритетного трафика приложений реального времени.

4.3.4. Технология Gigabit Ethernet

В 1995 г. группой производителей сетевого оборудования было образовано некоммерческое объединение Gigabit Ethernet Alliance, целью которого стала разработка стандарта на технологию, обобщающую достижения отдельных компаний в области Ethernet-преемственного высокоскоростного стандарта. Технология получила название Gigabit Ethernet, и в 1998 г. комитет IEEE принял её спецификацию в качестве стандарта IEEE 802.3z.

Технология Gigabit Ethernet представляет собой эволюционное развитие технологии Fast Ethernet. В данной технологии используются такой же формат кадра (за исключением длины кадра — все кадры с длиной меньше 512 байт расширяются до 512 байт), механизм доступа к среде CSMA/CD и топологию. Изменения (как и в технологии Fast Ethernet) произошли как на физическом уровне, так и на уровне MAC, в частности изменились аппаратная составляющая, физическое кодирование, параметры сети.

Технология Gigabit Ethernet может использовать в качестве среды передачи как витую пару, так и оптоволокно, причём по сравнению с Ethernet и Fast Ethernet меняется как количество используемых проводников (для витой пары), так и методы кодирования. При кодировании сигнала применяются методы NRZI и MLT-3, при физическом кодировании — 8B/10B.

В технологии Gigabit Ethernet (как и в Fast Ethernet) возможна как дуплексная (full-duplex mode), так и полудуплексная передача данных. В режиме full-duplex вместо CSMA/CD используется соединение P2P (точка–точка) и отсутствует понятие коллизий — каждый узел одновременно передаёт и принимает кадры данных. Работа в данном режиме возможна только при соединении сетевого адаптера с коммутатором или же при непосредственном соединении коммутаторов.

Физическое соединение

Физически и логически сети на базе технологии Gigabit Ethernet имеют топологию звезда.

В качестве физической среды может использоваться:

– витая пара:
  – 4 пары UTP CAT.5 (1000Base-T),
  – 2 пары STP (100Base-CX);
– мультимодовый оптоволоконный кабель с длиной волны светового сигнала 850 нм (1000Base-SX);

– мультимодовый оптоволоконный кабель с длиной волны светового сигнала 1300 нм (1000Base-LX);
– одномодовый оптический кабель (1000Base-LH).

Проблемы технологии Gigabit Ethernet и их решение

Проблемы технологии Gigabit Ethernet:
– обеспечение приемлемого диаметра сети для работы на разделяемой среде — в связи с ограничениями, накладываемыми методом CSMA/CD на длину кабеля, версия Gigabit Ethernet для разделяемой среды допускала бы длину сегмента всего 25 м;
– достижение битовой скорости 1 Гбит/с на оптическом кабеле — технология Fibre Channel, физический уровень которой был взят за основу для оптоволоконной версии Gigabit Ethernet, обеспечивает скорость передачи данных всего 800 Мбит/с;
– использование в качестве кабеля витой пары.

Для расширения максимального диаметра сети Gigabit Ethernet в полудуплексном режиме до 200 м был увеличен минимальный размер кадра — с 64 до 512 байт (без учёта преамбулы). Это повлекло за собой увеличение времени двойного оборота до 4095 bt, что сделало допустимым диаметр сети около 200 м при использовании одного повторителя.

Для увеличения длины кадра до требуемой величины сетевой адаптер должен дополнить поле данных до длины 448 байт так называемым расширением (extention), представляющим собой поле, заполненное запрещёнными символами кода 8B/10B, которые невозможно принять за коды данных.

Для сокращения накладных расходов при использовании слишком длинных кадров для передачи коротких квитанций узлам разрешено передавать несколько кадров подряд, без передачи среды другим станциям. Такой режим получил название *Burst Mode — форсированный режим передачи данных*. Станция может передать подряд несколько кадров с общей длиной не более 65 536 бит или 8192 байт. Если станции нужно передать несколько небольших кадров, то она может не дополнять их до размера в 512 байт, а передавать подряд до исчерпания предела в 8192 байт (в этот предел входят все байты кадра, в том числе преамбула, заголовок, данные и контрольная сумма).

Достоинства и недостатки технологии Gigabit Ethernet

Достоинства:
– увеличение пропускной способности сегментов сети до 1 Гбит/с;
– сохранение совместимости с методом случайного доступа CSMA/CD;
– сохранение формата кадра Ethernet;

– сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных — витой пары и оптоволоконного кабеля.

Недостатки (унаследованы от Ethernet):

– большие задержки доступа к среде при коэффициенте использования среды выше 30–40%, что связано с применением алгоритма доступа CSMA/CD;
– небольшие расстояния между узлами даже при использовании оптоволокна, что связано с работой метода обнаружения коллизий;
– отсутствие механизмов выбора резервных связей;
– отсутствие поддержки приоритетного трафика приложений реального времени.

4.4. Сети с маркерным доступом

4.4.1. Технология Token Bus

Схема передачи данных

Технология Token Bus определяет метод доступа к шине с передачей маркера. При инициализации сети станции образуют кольцо, и в соответствии с их адресами им присваиваются номера от старших к младшим. Инициализация кольца осуществляется следующим образом. В начальный момент станция включается и слушает канал. Если она не обнаруживает признаков передачи, то генерирует маркер. Если других станций не обнаружилось, то станция устанавливает кольцо из себя самой. Периодически станция генерирует специальные кадры, приглашая другие станции включиться в кольцо. Если в начальный момент сразу две станции были включены, то запускается алгоритм разрешения коллизий.

После процедуры инициализации кольца станция, имеющая наибольший номер, может послать первый кадр. Передача осуществляется в течение определённого промежутка времени, по истечении которого станция должна передать маркер следующей станции. Передача кадра разрешена только станции, владеющей маркером. Если у станции нет данных для передачи, то она передаёт маркер дальше. Коллизий в сети на базе Token Bus не возникает, так как по сети циркулирует только один маркер и только одна станция может передавать данные.

Следует отметить, что на порядок передач влияют только логические номера станций, а не их физическое размещение. Маркер передаётся только логическому соседу.

Схема приоритетов

Технология Token Bus определяет четыре приоритета для кадров: 0, 2, 4 и 6. Если маркер попадает на станцию с приоритетом 6 и у неё есть кадр на передачу, то она его передаёт. Если нет, то маркер передаётся станции с приоритетом 4. Эта подстанция передаёт свои кадры в течение своего интервала времени либо по истечении определённого промежутка передаёт маркер подстанции с приоритетом 2. Так продолжается до тех пор, пока подстанция с приоритетом 0 не перешлёт свои кадры или её таймер не исчерпается и она отдаст маркер следующей станции. Станция с наивысшим приоритетом используется для передачи трафика реального времени.

Поддержка логического кольца

Процедура поддержки логического кольца применяется при включении и выключении станций. После процедуры инициализации кольца интерфейс каждой станции хранит адреса предшествующей и последующей станций в кольце. Периодически станция, удерживающая маркер, рассылает специальный кадр, предлагая новым станциям присоединиться к кольцу. В этом кадре указаны адрес отправителя и адрес следующей за ним станции в кольце. Станции с адресами в этом диапазоне адресов могут присоединиться к кольцу. Таким образом сохраняется упорядоченность адресов в кольце. Если ни одна станция не откликнулась на посланный кадр, то станция, удерживающая маркер, закрывает окно ответа и продолжает функционировать в обычном режиме. Если есть ровно один отклик, то откликнувшаяся станция включается в кольцо и становится следующей в кольце. Если две или более станции откликнулись, то фиксируется коллизия, и станция, удерживающая маркер, запускает алгоритм разрешения коллизий.

Если станция решила отсоединиться от сети, то после получения маркера она посылает последующей станции специальный кадр, указывающий, что её предшественником будет станция, ранее предшествующая отсоединяющейся станции. После этого происходит отсоединение станции.

Физическое соединение

Физически шина с маркером имеет линейную или древовидную топологию. Логически станции объединены в кольцо.

В качестве физической среды используется 75-омный коаксиальный кабель или витая пара. Сеть способна обеспечить пропускную способность до 10 Мбит/с при полосе пропускания кабеля 12 МГц.

Формат блока данных

В сети Token Bus циркулируют два типа блока данных: *блоки маркеров* (рис. 4.9) и *блоки данных/команд* (рис. 4.10).

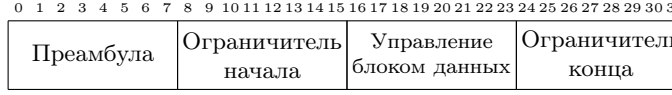


Рис. 4.9. Формат маркера Token Bus

1 байт	1 байт	1 байт	2 или 6 байт	2 или 6 байт	$\geqslant 0$ байт	4 байта	1 байт
Преамбула	Огр. начала	Упр. блоком данных	Адрес отправителя	Адрес получателя	Данные	Контр. сумма	Огр. конца

Рис. 4.10. Формат блока данных Token Bus

Поле *преамбула (Preamble)* (1 или более байт) предназначено для синхронизации таймера получателя.

Поле *ограничитель начала (Start Delimiter)* (длина 1 байт) указывает на начало маркера (или блока данных/команд), содержит сигнальные структуры, которые отличают его от остальной части блока данных.

Поле *управление блоком данных (Frame Control)* (длина 1 байт) указывает на размер адресных полей (2 или 6 байт), на тип кадра (маркер или управляющий/информационный), на приоритет кадров, может содержать подтверждения корректного или некорректного получения кадра, а также другую управляющую информацию (например, коды команд для включения станции в кольцо или исключения станции из кольца).

Поле *ограничителя конца (End Delimiter)* (длина 1 байт) содержит неинформационные символы, указывающие на конец маркера или блока данных/команд.

Поля *адрес отправителя (Source Address)* и *адрес получателя (Destination Address)* идентифицируют станции пункта назначения и источника, длина адресов может быть 6 байт или 2 байта.

Поле *данные (Data)* может иметь длину не более 8182 байт при 2-байтном адресе и 8174 байт при 6-байтном адресе, что в пять раз больше, чем в стандарте IEEE 802.3.

Поле *контрольная сумма (Frame Check Sequence)* содержит контрольную сумму, используемую для контроля ошибок. Если имеется повреждение, то блок данных отбрасывается.

Достоинства и недостатки

Достоинства:
- сеть может быть сконфигурирована для гарантированного пропускания определённого трафика, например цифрового голоса или мультимедиа;
- сеть имеет хорошую нагрузочную характеристику и неплохо работает при высоких нагрузках.

Следует отметить, что данная технология устарела и сейчас не используется, что и является её недостатком.

4.4.2. Технология Token Ring

Технология Token Ring разработана компанией IBM в 1970-х гг. Сети, построенные на базе Token Ring, были рассчитаны на скорость обмена 4 и 16 Мбит/с при числе сегментов до 250. IEEE в 1985 г. приняла данную технологию в качестве стандарта IEEE 802.5. При этом в стандарте IEEE 802.5 топология не оговорена, а сетевая среда не регламентирована.

Схема передачи данных

Станция может начать передачу данных только после получения от предыдущей станции специального кадра — маркера доступа.

Если станция готова к передаче данных, то

1. узел-отправитель:
   - ждёт получения маркера,
   - захватывает маркер (на определённое время, после истечения которого станция обязана завершить передачу своего очередного кадра и передать маркер доступа следующей станции),
   - меняет в маркере один бит, преобразующий маркер во флаг начала кадра, вносит в кадр информацию, подлежащую пересылке,
   - посылает кадр следующей станции[1];
2. переданный в сеть кадр будет двигаться по сети от станции к станции, пока не попадёт в узел, которому он адресован;
3. узел назначения:
   - копирует необходимую информацию,
   - устанавливает флаг копирования (FCI), подтверждающий успешную доставку кадра адресату,
   - возвращает кадр в сеть;

[1] Когда информационный блок циркулирует по кольцу, маркер в сети отсутствует. Поэтому другие станции, желающие передать информацию, вынуждены ожидать.

4. кадр продолжает движение по сети от станции к станции, пока не попадёт в узел-отправитель, где он будет уничтожен; путём контроля API (индикатора распознавания кадра адресатом) проверяется, подключена ли к сети станция назначения.

Система приоритетов

В кадре Token Ring за управление доступом отвечают два поля — *приоритет* и *резервирование*.

Станция может завладеть маркером только, если её приоритет равен или выше приоритета маркера. Если маркер уже захвачен и преобразован в информационный кадр, то только станция с приоритетом выше, чем у станции отправителя, может зарезервировать маркер на следующий цикл.

Станции, которые подняли приоритет маркера, должны его восстановить после завершения передачи.

Физическое соединение

Топологию сети Token Ring можно рассматривать с двух позиций:
- логически — кольцо,
- физически — звезда.

Отдельные станции присоединяются к сети через специальные концентраторы — *многостанционные устройства доступа (MultiStation Access Unit, MSAU)*, которые соединены между собой, образуя кольцо (рис. 4.12 и 4.11). MSAU может выполнять следующие функции: централизовывать задание конфигурации, отключать неисправные станции, контролировать работу сети и т.д. Для присоединения кабеля к MSAU применяются специальные разъёмы, которые обеспечивают замкнутость кольца даже при отключении абонента от сети. Кабель содержит в себе две разнонаправленные линии связи. В составе MSAU имеются шунтирующие реле для исключения станций из кольца.

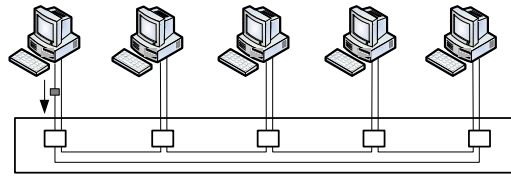


Рис. 4.11. Подсоединение узлов сети Token Ring через концентратор

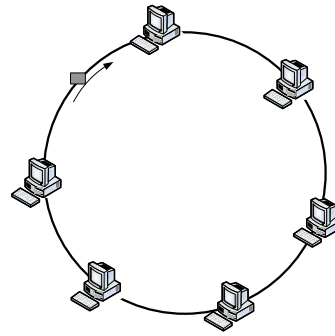


Рис. 4.12. Кольцо Token Ring

Механизмы обнаружения и предотвращения сетевых сбоев и ошибок

В сетях Token Ring существует несколько механизмов обнаружения и предотвращения сетевых сбоев и ошибок:

- присвоение одной из станций функций *активного монитора*, который играет роль центрального источника синхронизации для других станций сети, удаляет из кольца бесконечно циркулирующие кадры, генерирует новые кадры, осуществляет контроль работоспособности сети путём вывода из кольца станций, являющихся источником дефективных кадров;
- перепрограммирование MSAU для проверки наличия проблем и выборочного удаления при необходимости станций из кольца;
- применение «сигнализирующего» (*beaconing*) алгоритма:
  - станция, обнаружившая неисправность сети, высылает сигнальный блок данных, указывающий *домен неисправности*, состоящий из станции, сообщающей о неисправности, её ближайшего активного соседа, расположенного дальше по течению потока информации, и всего, что находится между ними;
  - сигнализация инициализирует *процесс автореконфигурации* (*autoreconfiguration*), в ходе которого узлы, расположенные в пределах отказавшего домена, автоматически выполняют диагностику, пытаясь реконфигурировать сеть вокруг отказавшей зоны.

Формат блока данных

В сетях на базе Token Ring циркулируют два типа блока данных: *блоки маркеров* (рис. 4.13) и *блоки данных/команд* (рис. 4.14).

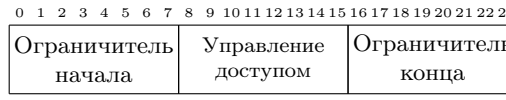


Рис. 4.13. Формат маркера Token Ring

1 байт	1 байт	1 байт	6 байт	6 байт	$\geqslant 0$ байт	4 байта	1 байт
Огр. начала	Упр. доступом	Упр. блоком данных	Адрес отправителя	Адрес получателя	Данные	Контр. сумма	Огр. конца

Рис. 4.14. Формат блока данных/команд Token Ring

Блок маркера имеет длину 3 байта. Блок данных и блок команд могут иметь разные размеры в зависимости от размеров информационного поля. Блоки данных переносят информацию для протоколов более высоких уровней, а блоки команд содержат управляющую информацию.

Поле *ограничитель начала (Start Delimiter)* (длина 1 байт) указывает на начало маркера (или блока данных/команд), содержит сигнальные структуры, которые отличают его от остальной части блока данных.

Поле *управление доступом (Access Control)* (длина 1 байт) содержит следующие поля:

- поле приоритета,
- поле резервирования,
- бит маркера, используемый для дифференциации маркера и блока данных/команд,
- бит монитора, используемый активным монитором для определения, циркулирует какой-либо блок в кольце непрерывно или нет.

Поле *ограничителя конца (End Delimiter)* (длина 1 байт) сигнализирует о конце маркера (или блока данных/ команд), содержит также бит для индикации повреждённого блока и бит идентификации блока, являющегося последним в логической последовательности.

Поле *управление блоком данных (Frame Control)* (длина 1 байт) указывает на тип содержимого блока — данные или управляющая информация. В управляющих блоках это поле определяет тип управляющей информации.

Поля *адрес отправителя (Source Address)* и *адрес получателя (Destination Address)* идентифицируют станции пункта назначения и источника. Для IEEE 802.5 длина адресов равна 6 байтам.

Поле *данные (Data)* содержит передаваемые данные. Длина этого

поля ограничена временем удержания маркера кольца.

Поле *контрольная сумма (Frame Check Sequence)* содержит контрольную сумму, зависящую от содержания блока данных, при помощи которой проверяется целостность кадра.

Применение

Сеть на базе технологии Token Ring может применяться для приложений, требующих предсказуемости задержки получения информации и высокой надёжности, например в сетях сопряжения с мейнфреймами.

Достоинства и недостатки

Достоинства:
- в сетях на базе технологии Token Ring не может быть коллизий, так как передавать информацию по сети может только одна станция, захватившая маркер, остальные станции вынуждены ожидать освобождения маркера;
- можно вычислить максимальное время, которое пройдёт, прежде чем любая станция сети сможет начать передачу данных.

Недостатки:
- технология Token Ring представляет собой проприетарный стандарт (IBM);
- технология Token Ring практически прекратила своё развитие;
- построение сетей на базе технологии Token Ring не получило распространения.

4.4.3. Технология FDDI

Сеть FDDI (Fiber Distributed Data Interface — волоконно-оптический распределённый интерфейс данных) представляет собой волоконно-оптическое маркерное кольцо со скоростью передачи данных 100 Мбит/с.

Стандарт FDDI был разработан комитетом X3T9.5 (впоследствии переименован в X3T12) ANSI в середине 1980-х гг. После завершения работы над FDDI ANSI представила его на рассмотрение в ISO. ISO разработала международный вариант FDDI, который полностью совместим с вариантом стандарта, разработанного ANSI.

Схема передачи данных

Двойное кольцо в сети FDDI рассматривается как общая разделяемая среда передачи данных, для которой в качестве метода доступа определён *метод маркерного кольца*, который близок к методу доступа сетей Token Ring.

Станция может начать передачу данных только после получения от предыдущей станции специального кадра — маркера доступа. Маркер — сигнал управления, состоящий из уникальной последовательности символов, которая циркулирует по кольцу после каждой информационной передачи. Если же в момент принятия маркера у станции нет данных для передачи по сети, то она немедленно передаёт маркер следующей станции.

Если станция готова к передаче данных, то

- узел-отправитель:
  - ждёт получения маркера,
  - захватывает маркер (на определённое время — *время удержания маркера (Token Holding Time, THT)*, после истечения которого станция обязана завершить передачу своего очередного кадра и передать маркер доступа следующей станции),
  - меняет в маркере один бит, преобразующий маркер во флаг начала кадра, вносит в кадр информацию, подлежащую пересылке, посылает кадр следующей станции[1];
- переданный в сеть кадр будет двигаться по сети от станции к станции, пока не попадёт в узел, которому он адресован;
- узел назначения:
  - копирует кадр в свой внутренний буфер,
  - проверяет корректность полученного кадра (в основном по контрольной сумме),
  - передаёт поле данных кадра для последующей обработки протоколу вышележащего уровня,
  - в исходном кадре отмечает следующие признаки: распознавание адреса, копирование кадра и отсутствие или наличие в нём ошибок,
  - возвращает кадр в сеть;
- вновь переданный в сеть кадр будет двигаться по сети от станции к станции, пока не попадёт в исходный узел-отправитель;
- узел-отправитель:
  - получив кадр, проверяет признаки кадра (получен ли кадр станцией назначения, был ли повреждён[2]),
  - удаляет кадр из сети,
  - передаёт маркер доступа следующей станции.

[1] Когда информационный блок циркулирует по кольцу, маркер в сети отсутствует. Поэтому другие станции, желающие передать информацию, вынуждены ожидать.

[2] Процесс восстановления информационных кадров осуществляют протоколы более высоких уровней.

Механизм адаптивного планирования нагрузки

В сетях на базе технологии FDDI вместо системы приоритетов и резервирования, используемой в сетях на базе технологии Token Ring, применяется механизм адаптивного планирования нагрузки.

Каждая станция сравнивает реальное *время обращения маркера по кольцу (Token Rotation Time, TRT)* с заранее установленным *контрольным временем прибытия маркера (Target Token Rotation Time, TTRT)*, после чего делается вывод о слабой или сильной загруженности сети. При слабой загрузке сети станция может использовать асинхронный режим передачи информации (т.е. осуществить передачу дополнительных данных независимо от других станций). При сильной загруженности сети станция может применять только синхронный режим передачи данных, при котором передача осуществляется лишь в течение выделенного времени.

Физическое соединение

Топологию сети, построенной на базе технологии FDDI, можно рассматривать с двух позиций:

- физически:
  - двойное кольцо без деревьев,
  - двойное кольцо с деревьями,
  - дерево;
- логически:
  - разделяемое кольцо.

При этом первичное кольцо используется для передачи данных, а вторичное кольцо является дублирующим (рис. 4.15).

Физически кольцо состоит из двух или более двухточечных соединений между смежными станциями. Трафик по кольцам движется в противоположных направлениях.

Оборудование сети:

- станции:
  - *станции двойного подключения (Dual-Attachment Stations, DAS)* — подключаются как к внутреннему, так и к внешнему кольцу сети,
  - *станции одинарного подключения (Single-Attachment Stations, SAS)* — подключаются только к внешнему кольцу сети и только через концентратор или обходной коммутатор, имеющий возможность отключить их при сбое;
- *связующие концентраторы (Wiring Concentrators)* — представляют собой точки подключения к сети, выполняют также функции управления, такие как контроль работы сети, диагностика неисправностей, реконфигурация сети; бывают двух типов:

  - *концентраторы двойного подключения (Dual-Attachment Concentrator, DAC)* — подключаются как к внутреннему, так и к внешнему кольцу сети,
  - *концентраторы одинарного подключения (Single-Attachment Concentrator, SAC)* — подключаются только к внешнему кольцу сети;
- *обходные коммутаторы (Bypass Switches)* — располагаются между станцией и кольцом и позволяют отключить станцию от сети при возникновении сбоев, замкнув сигнал на себя.

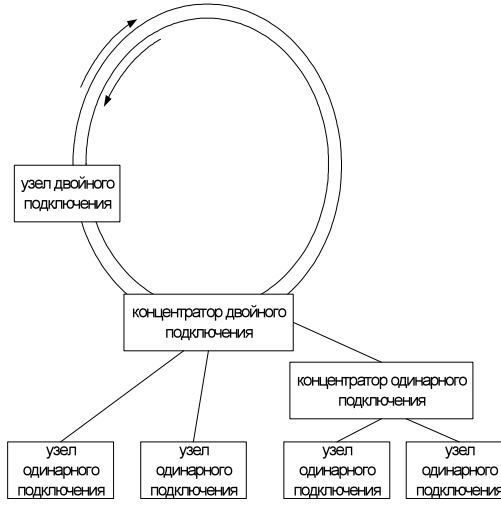


Рис. 4.15. Двойное кольцо FDDI

Основные параметры

Основные параметры сети FDDI:

- поддержка до 500 узлов с максимальным расстоянием между соседними узлами 2 км (45 км — если используется одномодовый оптоволоконный кабель)[1];
- максимальная длина кольца — 20 км (200 км, если используется одномодовый оптоволоконный кабель, по 100 км на кольцо)[2];

[1] Ограничение связано с затуханием сигнала в кабеле.

[2] Ограничение связано с необходимостью ограничения времени полного прохождения сигнала по кольцу для обеспечения предельно допустимого времени доступа.

– переменный размер кадра (до 4500 байт);
– длина волны — 1300 нанометров;
– максимальная скорость передачи — 100 МБод или 12.5 Мбит/с[1];
– реальная скорость работы — 80 МБод или 10 Мбит/с;
– рабочая частота — 125 МГц;
– основной вид кабеля — многомодовый или более качественный одномодовый *(Single Mode Fiber, SMF)*[2] оптоволоконный кабель,
– разъём — оптический разъём *MIC (Media Interface Connector)* (или разъём SMF-MIC для SMF-кабеля)[3], который обеспечивает подключение двух волокон кабеля, соединённых с вилкой MIC, к двум волокнам порта станции, соединённых с розеткой MIC;
– источник света — светодиоды (LED) или лазерные диоды с длиной волны $1,3$ мкм;
– метод кодирования сигнала — MLT-3;
– метод физического кодирования — 4B/5B.

Отказоустойчивость сетей на базе технологии FDDI

Основным способом обеспечения отказоустойчивости является подключение станций к двум кольцам. В нормальном режиме работы сети данные передаются по внешнему кольцу, а внутреннее кольцо при этом не используется. При возникновении сбоя в сети внешнее кольцо объединяется с внутренним, образуя таким образом единое кольцо. Данную операцию осуществляют концентраторы и/или сетевые адаптеры FDDI.

Другим способом обеспечения отказоустойчивости является использование различных процедур, определяющих наличие отказа в доступе к сети и производящих необходимую реконфигурацию. При единичном отказе сеть полностью восстанавливает свою работоспособность, а при множественных отказах сеть распадается на несколько несвязанных, но функционирующих сетей.

Ещё одним способом обеспечения отказоустойчивости является метод доступа к среде, т.е. использование метода маркерного кольца, который исключает возникновение коллизий и позволяет с высокой степенью вероятности просчитать время передачи маркера или данных.

---

[1] Бод — единица измерения скорости цифрового потока. Для некодированного цифрового сигнала 1 Бод=1 бит/с. Для кодирования с избыточностью — скорости разные. МБод — миллион сигналов в секунду.

[2] В этом случае дальность физического соединения между соседними узлами может увеличиться до 40–60 км в зависимости от качества кабеля, разъёмов и соединений.

[3] Кроме разъёмов MIC допускается использование разъёмов ST и SC.

Формат блока данных

В сетях FDDI циркулируют два типа блока данных: *маркеры* (рис. 4.16) и *блоки данных/команд* (рис. 4.17).

$\geqslant$ 2 байт	1 байт	1 байт	1 байт
Преамбула	Ограничитель начала	Управление блоком данных	Ограничитель конца

Рис. 4.16. Формат маркера FDDI

$\geqslant$ 2 байта	1 байт	1 байт	6 байт	байт	$\geqslant$ 0 байт	4 байта	$\geqslant$ 2 байта
Преамбула	Огр. начала	Упр. блоком данных	Адрес отправителя	Адрес получателя	Данные	Контр. сумма	Сост. фрейма

Рис. 4.17. Формат блока данных FDDI

Блок маркера без преамбулы имеет длину 3 байта. Блок данных и блок команд могут иметь разные размеры в зависимости от размеров информационного поля. Блоки данных переносят информацию для протоколов более высоких уровней, а блоки команд содержат управляющую информацию.

Поле *преамбула (Preamble)* (2 или более байт) используется для синхронизации. Первоначально имеет размер 8 байт, но станции, через которые проходит кадр, могут менять (уменьшать) её размер.

Поле *ограничитель начала (Start Delimiter)* (длина 1 байт) указывает на начало маркера (или блока данных/команд), содержит сигнальные структуры, которые отличают его от остальной части блока данных.

Поле *управление блоком данных (Frame Control)* (длина 1 байт) указывает на размер адресных полей (2 или 6 байт), на тип кадра (синхронный/асинхронный и управляющий/информационный), а также может содержать другую управляющую информацию (например, коды команд для управляющего кадра).

Поле *ограничителя конца (End Delimiter)* (длина 1 байт) содержит неинформационные символы, указывающие на конец маркера (или блока данных/команд).

Поля *адрес отправителя (Destination Address)* и *адрес получателя (Source Address)* идентифицируют станции пункта назначения и источника, длина адресов может быть 6 байт (по аналогии с Ethernet и

Token Ring) или 2 байта. При этом поле адреса назначения может содержать индивидуальный, групповой или широковещательный адрес, в то время как адрес источника идентифицирует только одну станцию, отправившую блок данных.

Поле *данные (Data)* (0 до 4478 байт) содержит либо информацию, предназначенную для протокола высшего уровня, либо управляющую информацию.

Поле *контрольная сумма (Frame Check Sequence)* содержит контрольную сумму, зависящую от содержания блока данных, при помощи которой проверяется целостность кадра. Если повреждение имеется, то блок данных отбрасывается.

Поле *состояния блока данных (Frame Status)* позволяет станции источника определять, не появилась ли ошибка и был ли блок данных признан и скопирован принимающей станцией.

Применение

Сеть на базе технологии FDDI может применяться в качестве надёжной высокоскоростной магистрали или высокопроизводительной сети многоцелевого назначения с большим числом узлов.

Достоинства и недостатки

Достоинства:

- надёжность:
  - обеспечение избыточности благодаря двойной кольцевой конфигурации сети,
  - возможность сохранения работоспособности сети при единичных и множественных обрывах посредством сегментирования участков сети;
- отказоустойчивость:
  - возможность двойного соединения (Dual Homing) станции с сетью FDDI (два порта станции подключаются к двум разным концентраторам) позволяет активировать резервную связь при возникновении сбоев,
  - реализация так называемого «оптического обхода» обеспечивает прохождение светового сигнала по сети при сбоях в питании станции — световой сигнал обойдёт неактивную станцию через оптический переключатель (Optical Bypass Switch),
  - однократный обрыв кабеля в любом месте кольца приведёт к активации второго волоконно-оптического кольца, так как станции, расположенные по обе стороны обрыва, переконфигурируют путь циркуляции маркера и данных;
- встроенное управление:

– каждый узел имеет объект управления, предоставляя большое число служб,
– есть возможность SNMP управления.

Недостатки:

– высокая цена, обусловленная дорогими трансиверами, преобразующими электрический сигнал в оптический, и наоборот.

4.5. Технология 100VG-AnyLAN

Технология 100VG-AnyLAN стала результатом проекта компаний AT&T и HP по разработке альтернативной Fast Ethernet-технологии — 100Base-VG со скоростью передачи данных 100 Мбит/с, использующей в одной сети как кадры формата Ethernet, так и кадры формата Token Ring. Для стандартизации технологии в 1993 г. фирмами IBM и HP был образован комитет IEEE 802.12, а в 1995 г. технология 100VG-AnyLAN получила статус стандарта IEEE 802.12.

4.5.1. Элементы сети 100VG-AnyLAN

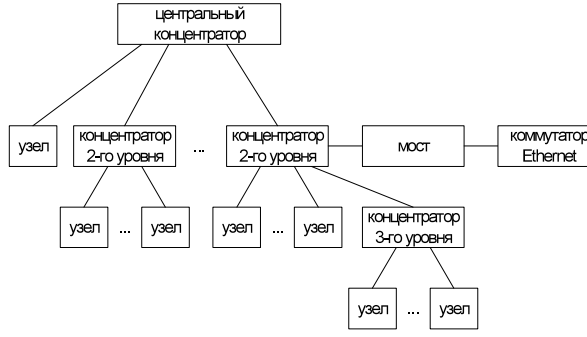


Рис. 4.18. Структура сети 100VG-AnyLAN

Сеть на базе технологии 100VG-AnyLAN (рис. 4.18) содержит следующие элементы:

– *узел* — компьютер или коммуникационное устройство технологии 100VG-AnyLAN (например, концентратор, коммутатор, мост, маршрутизатор);
– *центральный концентратор (или корневой концентратор)*:
  – управляет доступом к сети;
  – перенаправляет проходящие через него кадры узлу назначения;

- *концентратор 2-го, 3-го и т.д. уровня*:
  - имеет один порт (up-link) для присоединения в качестве узла к концентратору более высокого уровня и N портов (down-link) для присоединения узлов;
  - каждый порт может работать в одном из двух режимов: в *нормальном режиме*, когда порт передаёт только кадры, предназначенные узлу, подключённому к данному порту, или в *режиме монитора*, когда порт передаёт все кадры, обрабатываемые концентратором;
  - концентратор может быть настроен на работу или с форматом кадров Ethernet, или с форматом кадров Token Ring, причём концентраторы всей сети должны работать с пакетами только какого-нибудь одного формата;
- *мост, коммутатор или маршрутизатор* — применяются для соединения сетей 100VG-AnyLAN, использующих разные форматы кадров (кадры 802.3 Ethernet или кадры 802.5 Token Ring).

4.5.2. Схема передачи данных

Если узел сети готов к передаче данных, то

1) узел посылает концентратору, с которым он соединён, свой запрос на передачу;
2) концентратор циклически прослушивает все узлы по очереди и даёт на гарантированное время право передачи узлу, следующему по порядку за тем, который закончил передачу[1].

Узел в течение цикла кругового сканирования может передать через сеть только один кадр данных. Концентраторы, присоединённые как узлы к концентраторам верхних уровней иерархии, также выполняют свои циклы сканирования и передают запрос на передачу кадров концентратору. При этом N-портовый концентратор нижнего уровня может передать N кадров в течение цикла кругового сканирования.

4.5.3. Метод доступа Demand Priority

Метод *Demand Priority — приоритетный доступ по требованию* — представляет собой детерминированный метод разделения общей среды, использующий два уровня приоритетов: *низкий* — для обычных приложений и *высокий* — для мультимедийных приложений, чувствительных к задержкам.

Каждый концентратор имеет разные очереди для низкоприоритетных и высокоприоритетных запросов. Низкоприоритетные запросы

[1] Приоритет у узлов — географический, т.е. определяется номером порта нижнего уровня, к которому подключён узел.

обслуживаются до момента получения высокоприоритетного запроса.Чтобы перейти опять к обслуживанию низкоприоритетных кадров, должны быть обслужены все высокоприоритетные запросы. Чтобы обеспечить доступ для низкоприоритетных запросов в периоды высокой интенсивности поступления высокоприоритетных запросов, используется порог ожидания запроса. Если время ожидания низкоприоритетного запроса превышает этот порог, то ему присваивается более высокий приоритет.

4.5.4. Процедура подготовки к связи (Link Training)

Во время *процедуры подготовки к связи (Link Training)* концентратор и узлы обмениваются между собой управляющими пакетами специального формата. При этом проверяются правильность присоединения линий связи и их исправность, а также уровень ошибок. Одновременно концентратор получает информацию об особенностях узлов, подключённых к нему, их назначении и адресах, которые он заносит в таблицу, что позволяет ему перенаправлять полученные пакеты именно тем узлам, которым они адресованы. Концентраторы верхних уровней хранят таблицы адресов и тех узлов, которые подключены к концентраторам более низких уровней. Таким образом, основной (корневой) концентратор содержит в себе информацию о всех узлах сети.

Запускается данная процедура подготовки к связи узлом при включении питания или после подключения к концентратору, а также автоматически при большом уровне ошибок.

4.5.5. Основные параметры сети 100VG-AnyLAN

Ниже приведены основные параметры сети на базе технологии 100VG-AnyLAN:

- топология — звезда;
- типы физической среды:
  - 4-парная неэкранированная витая пара UTP CAT 3, 4, 5,
  - 2-парная неэкранированная витая пара UTP CAT 5,
  - 2-парная экранированная витая пара STP Type 1,
  - 2-парный многомодовый или одномодовый оптоволоконный кабель;
- максимальный диаметр сети — 8 км;
- максимальная длина сегмента:
  - UTP CAT 3,4 — 100 м,
  - UTP CAT 5 — 200 м,
  - STP Type 1 — 100 м,
  - оптоволокно — 2 км,
- кодирование сигнала — NRZ (для витой пары);

– физическое кодирование — 5B/6B;
– количество уровней каскадирования концентраторов — до 5;
– максимальное количество абонентов — 1024;
– рекомендуемое количество абонентов — до 250.

4.5.6. Достоинства и недостатки

Достоинства:
– высокая скорость передачи;
– централизованный метод управления обменом без конфликтов с гарантированием предельной величины времени доступа;
– совместимость на уровне форматов кадров с сетями Ethernet или Token Ring;
– кадры передаются не всем узлам сети, а только станции назначения, что затрудняет перехват сигнала.

Недостатки:
– не обладает полной совместимостью ни с одной из стандартных сетей;
– для совместимости с сетями Ethernet или с сетями Token Ring требуется дополнительное устройство — мост.

4.6. Технологии доступа с виртуальными каналами

4.6.1. Технология X.25

Технология X.25 разработана Международным консультативным комитетом по телефонии и телеграфии в 1976 г. для организации *глобальных сетей (Wide Area Network, WAN)* на базе *телефонных сетей общего пользования (ТфОП)*.

Стандарт X.25 описывает способы обмена информацией между удалёнными терминалами, локальными сетями и другими видами конечного оборудования. Стандарт предполагает обмен данными при помощи коммутации пакетов с установлением виртуальных соединений. Технология X.25 имеет свой стек протоколов с одноимённым названием, соответствующий трём нижним уровням модели ISO/OSI.

Принципы построения и компоненты сети X.25

Информационное взаимодействие в сети X.25 осуществляется на физическом, канальном и сетевом уровнях. На физическом уровне могут быть использованы любые универсальные или специализированные интерфейсы.

Компонентами сети являются устройства трёх основных категорий:

– *оконечное оборудование данных (Data Terminal Equipment, DTE)*,
– *оконечное оборудование канала передачи данных (Data Circuit-Terminating Equipment, DCE)*,
– *устройство коммутации пакетов PSE (Packet Switching Exchange)*.

Кроме того, в сети X.25 используют специальное устройство PAD (Packet Assembler/ Disassembler), предназначенное для обеспечения взаимодействия неспециализированных терминалов с сетью, для преобразования потока символов, который поступает от неспециализированного терминала в пакеты X.25, и выполнения обратного преобразования.

Для обеспечения информационного взаимодействия между компонентами сети X.25 применяется механизм организации виртуальных каналов. Между двумя терминалами устанавливается логическое виртуальное соединение на период обмена информацией, по завершении которого соединение разрывается. Существует также возможность установления постоянных виртуальных каналов.

Каждый терминал может одновременно устанавливать до 4096 виртуальных соединений или постоянных виртуальных каналов. Установленное соединение или постоянный виртуальный канал определяют маршрут движения пакетов при обмене информацией, а также скорость обмена. Скорости передачи в сетях на базе протокола X.25 определены и составляют: 1,2; 2,4; 4,8; 9,6; 19,2 Кбит/с — при использовании аналоговых абонентских линий; 64, 128, 192, 256, 384, 512, 768, 1024, 1536 и 2048 Кбит/с — при использовании выделенных линий и цифровых абонентских окончаний.

Обнаружение и коррекция ошибок

Технология X.25 разрабатывалась специально для передачи данных по линиям связи невысокого качества, в том числе по линиям ТфОП, где велика вероятность искажения информации. Поэтому основными задачами, поставленными при разработке технологии, стали:
– обеспечение передачи сообщений произвольного размера из произвольной комбинации бит;
– обеспечение выполнения процедур обнаружения ошибок на принимающей стороне;
– гарантия отсутствия дублирования и потерь компонентов (искажения) при возникновении ошибки во время передачи;
– обеспечение работы как двухточечных, так и многоточечных физических соединений;
– обеспечение подключения дуплексных и полудуплексных линий;
– обеспечение информационного обмена при значительных вариациях времени распространения сигнала.

Таким образом, одной из основных задач при разработке технологии стало обеспечение корректности принимаемых данных, для чего

был использован алгоритм обнаружения и коррекции ошибок. Его принцип состоит в вычислении контрольной суммы кадра передаваемой информации и сравнении принятых данных с полученным контрольным числом на приёме. В ответ на каждый принятый кадр данных получатель должен отправить источнику подтверждение, в котором отмечено о корректности переданной информации. Источник может передавать следующую порцию данных только после получения подтверждения. Использование такого алгоритма гарантирует защиту от ошибок, возникающих при передаче, но при этом требует обмена большим количеством служебной информации, что заметно снижает скорость обмена данными между двумя терминалами.

Структура блока данных

В сетях на базе технологии X.25 в качестве протокола канального уровня используется процедура *LAPB (Link Access Procedure Balanced)*. Рекомендация X.25 определяет два основных типа процедуры LAPB — *основной* и *расширенный*, отличающихся разрядностью счётчиков, которые используются для управления потоком кадров. На рис. 4.19 приведён формат кадра LAPB.

Рис. 4.19. Формат кадра LAPB

Поле *флаг (Flag)* (1 байт) — ограничивает блок данных LAPB. Протокол LAPB использует в качестве флага комбинацию из 8 бит, которая состоит из 6 единиц и двух нулей, обрамляющих эту последовательность (01111110). Процесс приёма кадра завершается при получении следующего флага. В том случае, если к моменту получения завершающего флага приёмник получил менее 32 бит, принятый кадр считается ошибочным и уничтожается.

Поле *адрес (Address)* (1 байт) — содержит бит C/R (Command/Response), указывающий, что включает блок данных — запрос или ответ. В зависимости от значения этого бита дальше следует физический адрес принимающей или передающей станции.

Поле *управление (Control)* (1 байт) — определяет тип кадра:

– информационный кадр (Information (I) frame) — содержит информацию более высоких уровней и определённую управляющую информацию: номера последовательностей кадров и бит P/F (Poll/Final), определяющий, является ли данный кадр последним в последовательности;

– управляющий кадр (Supervisory (S) frame) — содержит управляющую информацию и не содержит информационного поля, запрашивает и приостанавливает передачу, сообщает о состоянии канала и подтверждает приём информационных кадров;
– ненумерованный кадр (Unnumbered (U) frame) — предназначен для организации и разрыва логического соединения, согласования параметров линии и формирования сигналов о возникновении неустранимых ошибок в процессе передачи данных.

Информационное поле (Data) содержит данные более высоких уровней. Если кадр не является информационным, то данное поле отсутствует.

Поле *контрольная сумма (Frame Check Sequence — FCS)* (длина 2 байта) используется для обнаружения возможных ошибок при передаче.

Применение технологии X.25

Возможные сферы применения:
– обмен сообщениями между пользователями;
– обращение большого количества пользователей к удалённой базе данных, а также к удалённому хосту электронной почты;
– связь локальных сетей (при скоростях обмена не более 512 Кбит/с);
– объединение удалённых кассовых аппаратов и банкоматов.

Иными словами, технология X.25 применяется для организации сетей, в которых трафик не является равномерным во времени, а линия связи невысокого качества.

Достоинства и недостатки сети на базе технологии X.25

Достоинства:
– в режиме реального времени есть возможность разделять один и тот же физический канал между несколькими абонентами,
– передача данных может осуществляется по каналам телефонной сети общего пользования (выделенным и коммутируемым) оптимальным образом, т.е. с максимально возможной на указанных каналах скоростью и достоверностью передачи данных,
– возможно применение механизма альтернативной маршрутизации.

Недостатки:
– невозможность передавать такие виды информации, как голос и видео.

4.6.2. Технология Frame Relay

Frame Relay (FR) — ретрансляция кадров — технология доставки сообщений в сетях передачи данных с коммутацией пакетов.

В разработке стандартов Frame Relay приняли участие три организации:
- Frame Relay Forum (FRF) — международный консорциум, включающий в себя свыше 300 поставщиков оборудования и услуг, среди которых 3Com, Northern Telecom, Digital, Cisco, Netrix, Ascom Timeplex, Newbridge Networks, Zilog и др.;
- American National Standards Institute (ANSI) — Американский национальный институт по стандартизации;
- ITU-T (International Telecommunication Union) — Международный союз электросвязи.

В 1988 г. ITU-T (в то время CCITT) принял Рекомендацию I.122 «Обеспечение дополнительного пакетного режима», которая использовалась как часть серии стандартов ISDN. Комитет ANSI T1S1 занялся развитием положений I.122, завершившимся принятием стандартов, полностью определяющих Frame Relay. Стандарт T1.606 был одобрен в 1990 г., а остальные стандарты (T1.617, T1.618) приняты в 1991 г.

Принципы построения и компоненты сети Frame Relay

Физически сети Frame Relay образуют ячеистую структуру коммутаторов. Компоненты :
- *оконечное оборудование данных (Data Terminal Equipment, DTE)*;
- *оконечное оборудование каналов передачи данных (Data Circuit-terminating Equipment, DCE)*;
- *FR-адаптеры и FR-интерфейсы (FR assembler/disassembler, FRAD).*

Требования технологии Frame Relay:
- оконечные устройства должны поддерживать интеллектуальные протоколы более высоких уровней модели ISO/OSI;
- каналы связи должны быть свободны от ошибок;
- прикладные средства должны уметь осуществлять различные передачи.

Виртуальные каналы

Основу Frame Relay составляют *виртуальные каналы (Virtual Circuits)*. Виртуальный канал в сети Frame Relay представляет собой логическое соединение, которое создаётся между двумя устройствами DTE и используется для передачи данных.

В сети Frame Relay используется два типа виртуальных каналов — *коммутируемые (Switched Virtual Circuits, SVC)* и *постоянные (Permanent Virtual Circuits, PVC)*.

SVC устанавливается динамически. Для него стандарты передачи сигналов определяют, как узел должен устанавливать, поддерживать

и сбрасывать соединение. Процесс передачи данных с использованием SVC состоит из четырёх последовательных фаз:

- *установление вызова (Call Setup)* — создаётся виртуальное соединение между двумя DTE;
- *передача данных (Data Transfer)* — фаза непосредственной передачи данных;
- *ожидание (Idle)* — виртуальное соединение ещё существует, но передача данных через него уже не производится; если период ожидания превысит установленное значение тайм-аута, соединение может быть завершено автоматически;
- *завершение вызова (Call Termination)* — фаза завершения соединения.

PVC включает в себя конечные станции, среду передачи и все коммутаторы, расположенные между конечными станциями. После установки PVC для него резервируется определённая часть полосы пропускания, и двум конечным станциям не требуется устанавливать или сбрасывать соединение.

Процесс передачи данных по каналу PVC имеет всего две фазы:

- передача данных — фаза непосредственной передачи данных;
- ожидание — виртуальное соединение существует, однако передача данных через него не производится.

В отличие от SVC, постоянный канал PVC не может быть автоматически разорван в том случае, если он не используется для передачи данных.

PVC имеют два преимущества над SVC:

- могут обеспечить более высокую производительность, так как соединение устанавливается предварительно и впоследствии не разрывается;
- обеспечивают лучший контроль над сетью, так как провайдер или сетевой администратор может выбирать путь, по которому будут передаваться кадры.

Однако и SVC имеют ряд преимуществ над PVC:

- могут имитировать сети без установления соединений (необходимо, если пользователь использует приложение, которое не может работать в сети с установлением соединения);
- используют полосу пропускания только тогда, когда это необходимо (PVC должны постоянно её резервировать на тот случай, если она понадобится);
- требуют меньшей административной работы, поскольку устанавливаются автоматически, а не вручную.

Однако режим SVC не получил широкого распространения, в силу сложности в реализации. Как следствие, PVC является наиболее распространённым режимом связи в сети FR.

Формат блока данных

На рис. 4.20 приведён формат кадра Frame Relay.

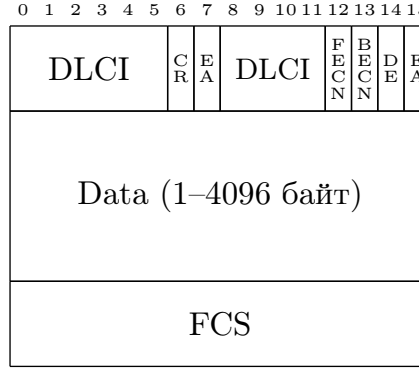


Рис. 4.20. Формат кадра Frame Relay

Поле *Флаг* обрамляет кадр Frame Relay.

Поле *Заголовок* содержит:

– поле *Идентификатор канала передачи данных (Data Link Connection Identifier, DLCI)* — определяет абонентский адрес в сети Frame Relay (стандарт FRF), состоит из шести бит первого октета и четырёх бит второго октета заголовка кадра (стандарты ANSI и ITU-T допускают размер заголовка до 4 байт);
– поле *Запрос/Ответ (Command/ Response, CR)* (2 бита) — зарезервировано для возможного применения в различных протоколах более высоких уровней модели ISO/OSI;
– бит *Расширение адреса (Extended Address, EA)* — устанавливается в конце каждого октета заголовка и указывает на наличие/отсутствие расширения заголовка Frame Relay на целое число дополнительных октетов с целью указания адреса, состоящего более чем из 10 бит, причём если бит имеет значение 1, то данный октет в заголовке последний;
– бит *Уведомление приёмника о явной перегрузке (Forward Explicit Congestion Notification, FECN)* — устанавливается аппаратурой канала данных в 1 для уведомления получателя сообщения о том, что произошла перегрузка в направлении передачи данного кадра;
– бит *Уведомление источника о явной перегрузке (Backward Explicit Congestion Notification, BECN)* — устанавливается аппаратурой канала данных в 1 для уведомления источника сообщения о том, что произошла перегрузка в направлении, обратном направлению

передачи содержащего этот бит кадра, после чего источник должен снизить интенсивность передаваемого потока данных;
- бит *Разрешения сброса (Discard Eligibility, DE)* — устанавливается в 1 (либо аппаратурой канала данных, либо оконечным оборудованием) в случае явной перегрузки и указывает на то, что данный кадр может быть уничтожен в первую очередь.

Информационное поле (Data) содержит данные пользователя и состоит из целого числа октетов. Его максимальный размер определён стандартом FRF и составляет 1600 байт (минимальный размер — 1 байт), но возможны и другие максимальные размеры (вплоть до 4096 байт). Содержание информационного поля пользователя передаётся без внесения изменений.

Поле *контрольная сумма (Frame Check Sequence, FCS)* (длина 2 байта) используется для обнаружения возможных ошибок при передаче. Содержит 16-разрядную контрольную сумму всех полей кадра Frame Relay, за исключением поля *Флаг*.

Адресация в сетях Frame Relay

Для идентификации виртуальных каналов в сети Frame Relay используется DLCI, который определяет номер виртуального порта для процесса пользователя. Обычно идентификатор DLCI имеет только локальное значение и не является уникальным в пределах сети. Конкретные значения DLCI для каждого пользователя определяются провайдером сервиса Frame Relay.

Дополнение в виде глобальной адресации позволяет применять идентификаторы узлов. При использовании этого дополнения значения, вставленные в поле DLCI блока данных, являются глобально значимыми адресами индивидуальных устройств конечного пользователя (например, маршрутизаторов). Аппаратура канала данных обязана обладать способностью определения принадлежности проходящего кадра конкретному PVC. Внутри сети Frame Relay могут использоваться различные сетевые адреса. Для разных интерфейсов одно и то же значение DLCI может применяться многократно.

Отличия протокола Frame Relay от HDLC

Отличия протокола Frame Relay от HDLC состоят в следующем:
- Frame Relay не предусматривает передачу управляющих сообщений;
- для передачи служебной информации используется специально выделенный канал сигнализации;
- отсутствует нумерация последовательно передаваемых (принимаемых) кадров, так как протокол Frame Relay не имеет никаких механизмов для подтверждения правильно принятых кадров.

Применение технологии Frame Relay

Данная технология применяется как для управления пульсирующим трафиком между локальными сетями и территориальной сетью, так и для передачи голоса.

Достоинства и недостатки

Достоинства:
- малое время задержки;
- простой формат кадров, содержащих минимум управляющей информации, следствием чего является высокая эффективность передачи данных (в предположении, что канал надёжен);
- независимость от протоколов верхних уровней модели ISO/OSI;
- предсказуемая пропускная способность;
- возможность контроля работоспособности (нагруженности) канала;
- возможность приоритезации разнородного трафика (для каждого типа трафика можно организовать своё виртуальное соединение).

Недостатки:
- Frame Relay не различает протоколы вышележащих уровней и, следовательно, нельзя приоритезировать трафик без организации дополнительных виртуальных соединений, что несёт дополнительные накладные расходы;
- отсутствие широковещательного множественного доступа;
- нет встроенных функций контроля доставки и управления потоком кадров (функции управления потоком выполняются протоколами верхних уровней).

4.7. Технологии региональных сетей

Региональные сети строятся по принципу функционального разделения по уровням доступа: опорная сеть (магистраль), уровень распределения/агрегации, уровень доступа (клиентский доступ).

4.7.1. Технологии опорной сети

Опорная сеть обычно имеет кольцевую топологию, обеспечивающую резервирование и повышенную надёжность. В качестве физической среды передачи данных применяется оптоволокно. Базовыми магистральными технологиями являются SONET/SDH, ATM, POS (Pocket over Sonet), EoSDH (Ethernet over SDH), DWDM, CWDM, DPT/RPR, Fast/Gigabit/10 Gigabit Ethernet.

На уровне доступа применяются следующие технологии: Ethernet (Ethernet, Fast Ethernet, Gigabit Ethernet), LRE, xDSL (HDSL, ADSL,

VDSL, SDSL), PNA (Phoneline Networking Alliance), Wireless (802.11), Infrared, PON (Passive Optical Network), EFM (Ethernet in the First Mile, IEEE 802.3ah), Satellite.

Для обеспечения повышенной надёжности и резервирования широко используется топологическая модель кольца. Кольца обычно создают на уровнях опорной сети и доступа.

SONET/SDH

Изначально основной задачей телекоммуникационных структур являлась передача голосового трафика. Скорость передачи данных задаётся относительно звука с *импульсной модуляцией (Pulse Code Modulation, PCM)* с частотой дискретизации 8 кГц и 8-битной дискретизацией. В результате получается *базовый поток (Digital Signal, DS0)* 64 Кбит/с. Потоки агрегируются и передаются по высокоскоростным каналам. Агрегирование происходит по *технологии временного мультиплексирования каналов (Time Division Multiplexing, TDM)*. Непосредственное слияние и разделение каналов производят специальные устройства — *мультиплексоры*. Например, на вход мультиплексора может поступать 30 потоков DS0 (64 Кбит/с $\times$ 30 + 2 сигнальных по 64 Кбит/с), а на выходе получается один E1 (2048 Кбит/с).

В свою очередь, для мультиплексирования потоков информации при формировании мощных региональных и межрегиональных каналов были разработаны стандарты для высокоскоростных оптических сетей связи — сначала *плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH)*, а затем и более совершенная *синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH)*, распространённая в Европе, и её американский аналог *SONET*.

SONET/SDH предполагает использование метода временно́го мультиплексирования и синхронизацию временны́х интервалов трафика между элементами сети и определяет уровни скоростей прохождения данных и физические параметры. Основными устройствами являются мультиплексоры, а физической средой передачи — оптоволокно. При построении сети SDH обычно используется топология двойного кольца. По одному кольцу передаётся синхронизирующая информация, а по другому — непосредственно трафик. Использование колец даёт возможность автоматического восстановления при авариях. Метод передачи — коммутирование каналов.

К достоинствам SONET/SDH относят:

- стандартизованность,
- масштабируемость,
- высокую надёжность (время восстановления порядка 50 мс).

К недостаткам SONET/SDH относят:

- ориентацию на передачу голосового трафика,

– фиксированную полосу пропускания, не зависящую от загрузки каналов,
– неэффективное использование колец.

SONET/SDH является самой зрелой и поэтому самой распространённой на данный момент технологией для построения магистральных каналов передачи данных. Основная область её применения — первичные сети операторов связи. Мультиплексоры, объединённые оптическими линиями связи, образуют единую среду, в которой прокладываются цифровые каналы между оборудованием телефонных сетей или сетей передачи данных. Кроме того, технология SONET/SDH может являться транспортной основой для более современных протоколов, таких как ATM, POS и MPLS.

ATM

Как решение проблемы создания мультисервисной и высокоскоростной технологии передачи данных была предложена *технология асинхронной передачи данных (Asynchronous Transfer Mode, ATM)*. В локальных сетях ATM распространения не получила, но до сих пор применяется при построении магистральных сетей. ATM может работать поверх SONET/SDH.

Технология ATM представляет собой транспортный механизм коммутации ячеек небольшого размера фиксированной длины (53 байта). Наиболее распространённая среда передачи для ATM — оптоволокно.

В ATM при соединении создаётся виртуальный канал. Далее коммутация ячеек происходит на основе идентификаторов виртуального канала (VPI/VCI), присутствующих в заголовках.

ATM имеет встроенную поддержку обеспечения гарантированного качества обслуживания.

POS

Для решения проблемы накладных расходов в случае передачи IP-трафика через сети SONET/SDH с использованием ATM была разработана технология *POS (Packet Over Sonet/SDH)*, непосредственно инкапсулирующая данные в кадры SDH. Практически получается интерфейс с IP-адресом, который использует все преимущества транспортной оптической технологии, не задействуя никаких промежуточных протоколов.

EoSDH

Отвечая потребностям рынка по передаче непосредственно Ethernet трафика по наследованным оптическим сетям, появилась технология

Ethernet over SONET/ SDH. Вначале допускались только соединения типа точка-точка, затем возникли и многоточечные каналы.

WDM

Непрерывно возрастающие объёмы трафика требуют повышения пропускной способности оптических магистралей. Кроме тривиального повышения скоростей передачи существует и другой способ решения данной задачи — уплотнение (мультиплексирование) каналов. Наиболее развитой в настоящее время является *технология оптического спектрального уплотнения*, называемая обычно *мультиплексированием с разделением по длине волны (Wavelength Division Multiplexing, WDM)*.

Принцип работы WDM следующий. Потоки данных от отдельных источников переносятся световыми волнами разной длины (каждому каналу принадлежит своя длина) и объединяются мультиплексором в единый многочастотный сигнал, который передаётся по оптическому волокну. На стороне приёмника происходит обратное преобразование.

Технология WDM соответствует физическому уровню сетевых взаимодействий и работает независимо от типа и формата передаваемых данных, то есть является протокольно-независимой. К WDM мультиплексору можно подключить практически любое оборудование: SONET/SDH, ATM, Ethernet.

WDM бывает двух видов: *плотное волновое мультиплексирование (Dense Wavelength Division, DWDM)* и *грубое волновое мультиплексирование (Coarse Wavelength Division, CWDM)*.

DWDM может обеспечить большое число спектральных каналов на одно оптоволокно (32, 64 или даже 128). Отсюда её основная отличительная особенность — малые расстояния между мультиплексными каналами.

CWDM-системы рассчитаны на меньшее число каналов (4, 8 или 16). Поэтому в них спектры соседних информационных каналов расположены на гораздо больших расстояниях друг от друга, чем в DWDM. Скорости передачи CWDM систем ниже, чем у DWDM.

DPT/RPR

Стандарт IEEE 802.17 (вобравший в себя DPT/RPR) позиционируется как высокоскоростная технология динамической передачи IP-пакетов, предназначенная для решения задач построения региональных сетей.

В DPT/RPR (IEEE 802.17) к IP-пакету добавляется прослойка второго уровня (MAC), пакет помещается в произвольную физическую оптическую среду (SONET/ SDH, WDM) с топологией двойного

кольца. Данные одновременно передаются по двум кольцам в противоположных направлениях. Поток данных в каждом кольце включает непосредственно транспортируемые в данном кольце данные и управляющие пакеты для соседнего кольца.

Достоинства:

- пакетно-ориентирован;
- не требуется дополнительная прослойка типа ATM для доступа к физической оптической среде;
- заложен высокий уровень резервирования и быстрая восстановимость в случае аварий (50 мс);
- эффективно используется ёмкость оптических каналов за счёт смешения контрольных и передаваемых данных.

4.7.2. Технологии уровня доступа

Существует широкий спектр решений для обеспечения абонентского доступа (так называемая «первая/последняя миля»): Ethernet (Ethernet, Fast Ethernet, Gigabit Ethernet), LRE, xDSL (HDSL, ADSL, VDSL, SDSL), PNA (Phoneline Networking Alliance), Wireless (802.11), Infrared, PON (Passive Optical Network), EFM (Ethernet in the First Mile alliance 802.3ah), Satellite.

VLAN

Для построения развитых Ethernet сетей используют *технологию виртуальных локальных сетей (Virtual Private Lan, VLAN)* (IEEE 802.1Q), которая позволяет создавать в едином Ethernet-сегменте независимые логические области, ограничивающие на канальном уровне распространение трафика (в том числе и широковещательного). В заголовок Ethernet-фрейма вводится дополнительная информация о принадлежности к влану (VLAN); получается помеченный кадр данных (Tagged Vlan), который передаётся по транковым линиям (802.1Q Trunk). Это позволяет передавать по одному каналу данные нескольких изолированных локальных сетей. Дальнейшая коммутация происходит с учётом 802.1Q-метки. На выходе из коммутатора (например, на стороне клиентского порта) метка (Tag) убирается, что называется *вхождением порта в нетагированный влан (Untagged Vlan)*.

Обычно клиентские подсети изолируются друг от друга путём подключения к раздельным вланам (через порты с Untagged Vlan), а связь между ними организуется при помощи маршрутизатора через 802.1Q транки.

На практике использование вланов даёт возможность гибко изменять логическую организацию сети независимо от реальной физической топологии.

Q-in-Q

Непосредственным решением присущих 802.1Q вланам ограничений (например, их максимальное число 4096) явилась технология Q-in-Q. Операторское устройство, получающее клиентский кадр Ethernet, добавляет ещё одну 802.1Q-метку, которая и принимается во внимание при дальнейшей коммутации. Так получается целый блок меток, а сам процесс называется *стекированием вланов (802.1Q stacking)*. На выходе из провайдерской сети дополнительная метка удаляется. Это позволяет строить полностью прозрачные на канальном уровне сети.

STP

В сетях Ethernet коммутаторы поддерживают только древовидные связи (ациклический граф). Отказоустойчивость требует наличия петель (циклический граф). Технология STP (Spanning Tree Protocol) позволяет совместить оба требования.

После активирования коммутаторы обмениваются специальными информационными пакетами (BPDU), с помощью которых вначале выбирается корневой мост (который будет в итоге находиться на вершине древовидной структуры), а затем кратчайшие (в смысле пропускной способности) пути от каждого из коммутаторов до корневого. В конечном итоге формируется логическая беспетельная топология путём блокирования некоторых избыточных связей.

Расширением STP является стандарт *RSTP (Rapid Spanning Tree Protocol)*.

4.7.3. Технология Metro Ethernet

Преимущества Ethernet: высокая скорость, лёгкость масштабирования технологии, простота для массового использования.

Первоначально Ethernet строилась на базе разделяемой среды передачи, но позднее был введён коммутируемый Ethernet. Были созданы механизмы, гарантирующие качество обслуживания, что дало возможность использовать Ethernet для передачи мультимедийных данных.

Развитием технологии Ethernet для региональных сетей занимается *Metro Ethernet Forum (MEF)* — некоммерческая организация, созданная для продвижения концепции построения операторских сетей на основе Ethernet и ускорения их развёртывания во всём мире. В октябре 2003 г. форум Metro Ethernet ратифицировал первый стандарт, описывающий службы Metro Ethernet: MEF Technical Specification—Ethernet Services Model Phase 1.

По сравнению с технологиями, имеющими схожие потребительские свойства, например SDH/SONET, реализация Metro Ethernet обходится в среднем в 2–3 раза дешевле. В настоящее время все серьёзные

поставщики оборудования выпускают оборудование для Metro Ethernet и ведут активную маркетинговую политику по его продвижению на рынке.

Форум Metro Ethernet предложил модель услуг Metro Ethernet. В основе базовой модели лежит *городская Ethernet-сеть (Metro Ethernet Network, MEN)*, принадлежащая провайдеру. *Клиентское оборудование (Customer Equipment, CE)* подключается к сети с помощью интерфейса UNI (User Network Interface), который представляет собой стандартный Ethernet.

Для потребителя существует только Ethernet-интерфейс (UNI), которым он подключается к провайдеру услуг. Транспортные технологии, обеспечивающие работу Metro Network, для него скрыты.

Ключевым элементом модели является *виртуальное соединение Ethernet (Ethernet Virtual Connection, EVC)*, которе определяется как соединение двух и более UNI. По ним проходят данные в виде кадров Ethernet. EVC выполняет две функции:

– соединяет UNI потребителей и пропускает между ними Ethernet-фреймы; обеспечивает защищённость и безопасность;
– доставка кадров Ethernet производится с неизменяемыми параметрами: MAC-адреса и содержимое не изменяются в отличие от маршрутизирующих сетей.

MEF определяет два типа EVC: *один-к-одному (Point-to-Point)* и *многие-ко-многим (Multipoint-to-Multipoint)*.

MEF определяет два типа базовых услуг Ethernet: *E-Line (Ethernet Line service type)* и *E-LAN (Ethernet LAN service type)*:

– E-Line обеспечивает соединения point-to-point (аналог физических выделенных каналов или виртуальных выделенных каналов Frame Relay);
– E-LAN поддерживает multipoint соединения (подобен услуге прозрачных локальных сетей (TLS)).

Для полного определения сервисов провайдер услуг должен обозначить кроме типа сервиса (E-Line или E-LAN) на основе EVC ещё и атрибуты, которые можно сгруппировать по категориям:

– *физический интерфейс (Ethernet Physical Interface)* определяет параметры физического уровня модели OSI;
– *параметры трафика (Traffic Parameters)* определяют полосу пропускания;
– *дополнительные параметры качества трафика (Perfomance Parameters)*: доступность (Availability), задержка (Delay), джиттер (Jitter), потери (Loss);
– *классы обслуживания (Class of Service)*;
– *необходимость доставки служебных пакетов (Service Frame Delivery)*;
– *поддержка VLAN (Vlan Tag Support)*: 802.1q, Q-in-Q, MAC-in-MAC;

– *фильтры (Security Filters)*: разнообразная фильтрация фреймов на основе различных критериев;
– *мультиплексирование виртуальных соединений (Service multiplexing)*: поддержка нескольких EVC на одном UNI;
– *неизменность клиентских VLAN (Vlan Transparency)*: неизменность клиентских вланов CE-Vlan при переходе через UNI, т.е. входной CE-Vlan и выходной CE-Vlan для одного и того же EVC одни и те же;
– *связывание (Bundling)*: отображение нескольких вланов CE-Vlan на одно EVC (используя Q-in-Q).

4.8. Технологии беспроводного доступа

4.8.1. Методы доступа к среде в беспроводных сетях

Существует несколько базовых методов доступа (их ещё называют методами уплотнения или мультиплексирования), основанных на разделении между станциями таких параметров, как пространство, время, частота и код. Задача уплотнения — выделить каждому каналу связи пространство, время, частоту и/или код с минимумом взаимных помех и максимальным использованием характеристик передающей среды.

Уплотнение с пространственным разделением

Основано на разделении сигналов в пространстве, когда передатчик посылает сигнал, используя код c, время t и частоту f области s_i. То есть каждое беспроводное устройство может вести передачу данных только в границах определённой территории, на которой любому другому устройству запрещено передавать свои сообщения.

Уплотнение с частотным разделением (Frequency Division Multiplexing, FDM)

Каждое устройство работает на определённой частоте, благодаря чему несколько устройств могут вести передачу данных на одной территории. Это один из наиболее известных методов, так или иначе используемый в самых современных системах беспроводной связи.

Эта схема приводит к неоправданному расточительству частотных ресурсов, поскольку требует выделения своей частоты для каждого беспроводного устройства.

Уплотнение с временны́м разделением (Time Division Multiplexing, TDM)

В данной схеме распределение каналов идёт по времени, т. е. каждый передатчик транслирует сигнал на одной и той же частоте, но в различные промежутки времени (как правило, циклически повторяющиеся) при строгих требованиях к синхронизации процесса передачи.

Временны́е интервалы могут динамично перераспределяться между устройствами сети. Устройствам с большим трафиком назначаются более длительные интервалы, чем устройствам с меньшим объёмом трафика.

Основной недостаток систем с временны́м уплотнением — мгновенная потеря информации при срыве синхронизации в канале, например из-за сильных помех, случайных или преднамеренных.

Уплотнение с кодовым разделением (Code Division Multiplexing, CDM)

В данной схеме все передатчики транслируют сигналы на одной и той же частоте.

В схеме CDM каждый передатчик заменяет каждый бит исходного потока данных на CDM-символ — кодовую последовательность длиной в 11, 16, 32, 64 и т. п. бит (так называемый чип). Кодовая последовательность уникальна для каждого передатчика.

Приёмник знает CDM-код передатчика, сигналы которого должен воспринимать. Он постоянно принимает все сигналы и оцифровывает их. Затем в специальном устройстве (корреляторе) производится операция свёртки (умножения с накоплением) входного оцифрованного сигнала с известным ему CDM-кодом и его инверсией. В несколько упрощённом виде это выглядит как операция скалярного произведения вектора входного сигнала и вектора с CDM-кодом. Если сигнал на выходе коррелятора превышает некий установленный пороговый уровень, приёмник считает, что принял 1 или 0. Для увеличения вероятности приёма передатчик может повторять посылку каждого бита несколько раз. При этом сигналы других передатчиков с другими CDM-кодами приёмник воспринимает как аддитивный шум. Благодаря большой избыточности мощность принимаемого сигнала может быть сопоставима с интегральной мощностью шума. Сходства CDM-сигналов со случайным (гауссовым) шумом добиваются, используя CDM-коды, порождённые генератором псевдослучайных последовательностей. Этот метод также называется *методом расширения спектра сигнала посредством прямой последовательности (Direct Sequence Spread Spectrum, DSSS)*.

Наиболее сильная сторона данного уплотнения заключается в повышенной защищённости и скрытности передачи данных: не зная

кода, невозможно получить сигнал, а в ряде случаев и обнаружить его присутствие. Кроме того, кодовое пространство более значительно по сравнению с частотной схемой уплотнения, что позволяет без особых проблем присваивать каждому передатчику свой индивидуальный код.

Механизм мультиплексирования посредством ортогональных несущих частот (Orthogonal Frequency Division Multiplexing, OFDM)

Весь доступный частотный диапазон разбивается на достаточно много поднесущих (от нескольких сот до тысяч). Одному каналу связи (приёмнику и передатчику) назначают для передачи несколько таких несущих, выбранных из множества по определённому закону. Передача ведётся одновременно по всем поднесущим, т.е. в каждом передатчике исходящий поток данных разбивается на N субпотоков, где N — число поднесущих, назначенных данному передатчику. Распределение поднесущих в ходе работы может динамически изменяться.

Преимущества:

- Селективному замиранию будут подвержены только некоторые подканалы, а не весь сигнал. Если поток данных защищён кодом прямого исправления ошибок, то с этим замиранием легко бороться.
- OFDM позволяет подавить межсимвольную интерференцию. Межсимвольная интерференция оказывает значительное влияние при высоких скоростях передачи данных, так как расстояние между битами (или символами) мало. В схеме OFDM скорость передачи данных уменьшается в N раз, что позволяет увеличить время передачи символа в N раз. Таким образом, если время передачи символа для исходного потока составляет T_s, то период сигнала OFDM будет равен NT_s. Это позволяет существенно снизить влияние межсимвольных помех. При проектировании системы N выбирается таким образом, чтобы величина NT_s значительно превышала среднеквадратичный разброс задержек канала.

4.8.2. Стек протоколов IEEE 802.11 (WiFi)

Стандарты IEEE 802.11 (WiFi, Wi-Fi, Wireless Fidelity)[1] описывают *беспроводную технологию локальных сетей (Wireless Local Area Network, WLAN).*

Сети WLAN имеют ряд преимуществ перед обычными кабельными сетями:

- их можно быстро развернуть;

[1] По аналогии с Hi-Fi.

– пользователи мобильных устройств при подключении к локальным беспроводным сетям могут легко перемещаться в рамках действующих зон сети;
– скорость современных сетей довольно высока (до 108 Мбит/с), что позволяет использовать их для решения очень широкого спектра задач;
– может оказаться единственным выходом, если невозможна прокладка кабеля для обычной сети.

Вместе с тем беспроводные сети имеют ряд ограничений:
– меньшая, чем в проводных сетях, скорость;
– подверженность влиянию помех;
– более сложная схема обеспечения безопасности передаваемой информации.

На физическом уровне существует несколько вариантов спецификаций, которые отличаются используемым частотным диапазоном, методом кодирования и как следствие — скоростью передачи данных. Все варианты физического уровня работают с одним и тем же алгоритмом уровня MAC, но некоторые временные параметры уровня MAC зависят от используемого физического уровня.

Уровень доступа к среде стандарта 802.11

В сетях 802.11 уровень MAC обеспечивает два режима доступа к разделяемой среде:
– *распределённый режим (Distributed Coordination Function, DCF)*;
– *централизованный режим (Point Coordination Function, PCF)*.

Распределённый режим доступа DCF. В этом режиме реализуется *метод множественного доступа с контролем несущей и предотвращением коллизий (Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA)*. Вместо неэффективного в беспроводных сетях прямого распознавания коллизий по методу CSMA/CD здесь используется их косвенное выявление. Для этого каждый переданный кадр должен подтверждаться кадром положительной квитанции, посылаемым станцией назначения. Если по истечении оговорённого времени квитанция не поступает, станция-отправитель считает, что произошла коллизия.

Режим доступа DCF требует синхронизации станций. В спецификации 802.11 временны́е интервалы начинают отсчитываться от момента окончания передачи очередного кадра. Это не требует передачи каких-либо специальных синхронизирующих сигналов и не ограничивает величину пакета размером слота, так как слоты учитываются только при принятии решения о начале передачи кадра. Станция, которая хочет передать кадр, обязана предварительно прослушать среду.

Предусматривается два механизма обнаружения несущей: *физический* и *виртуальный*. Первый механизм реализован на физическом

уровне и сводится к определению уровня сигнала в антенне и сравнению его с пороговой величиной. Виртуальный механизм обнаружения несущей основан на том, что в передаваемых кадрах данных, а также в управляющих кадрах ACK и RTS/CTS содержится информация о времени, необходимом для передачи пакета (или группы пакетов) и получения подтверждения. Все устройства сети получают информацию о текущей передаче и могут определить, сколько времени канал будет занят, т.е. устройство при установлении связи сообщает всем, на какое время оно резервирует канал. Как только станция фиксирует окончание передачи кадра, она обязана отсчитать интервал времени, равный межкадровому интервалу (IFS). Если после истечения IFS среда все ещё свободна, начинается отсчёт слотов фиксированной длительности. Кадр можно передавать только в начале какого-либо из слотов при условии, что среда свободна. Станция выбирает для передачи слот на основании усечённого экспоненциального двоичного алгоритма отсрочки, аналогичного используемому в методе CSMA/CD. Номер слота выбирается как случайное целое число, равномерно распределённое в интервале [0, CW], где CW (Contention Window) — *конкурентное окно*.

Если же в начале какого-нибудь слота среда оказывается занятой, то вычитания единицы не происходит, и таймер «замораживается». В этом случае станция начинает новый цикл доступа к среде, изменяя только алгоритм выбора слота для передачи. Как и в предыдущем цикле, станция следит за средой и при её освобождении делает паузу в течение межкадрового интервала. Если среда осталась свободной, то станция использует значение «замороженного» таймера в качестве номера слота и выполняет описанную выше процедуру проверки свободных слотов с вычитанием единиц, начиная с замороженного значения таймера отсрочки.

Размер слота зависит от способа кодирования сигнала (например, для метода FHSS размер слота равен 28 мкс; для метода DSSS — 1 мкс). Размер слота выбирается таким образом, чтобы он превосходил время распространения сигнала между любыми двумя станциями сети плюс время, затрачиваемое станцией на распознавание занятости среды. Если такое условие соблюдается, то каждая станция сети сумеет правильно распознать начало передачи кадра при прослушивании слотов, предшествующих выбранному ею для передачи слоту. Это, в свою очередь, означает, что коллизия может иметь место только в том случае, когда несколько станций выбирают один и тот же слот для передачи. В этом случае кадры искажаются, и квитанции от станций назначения не приходят. Не получив в течение определённого времени квитанцию, отправители фиксируют факт коллизии и пытаются передать свои кадры снова. При каждой повторной неудачной попытке передачи кадра интервал [0, CW], из которого выбирается номер слота, удваивается. Если, например, начальный размер окна выбран равным 8 (т.е.

CW = 7), то после первой коллизии размер окна должен быть равен 16 (CW = 15), после второй последовательной коллизии — 32 и т.д. Начальное значение CW, в соответствии со стандартом 802.11, должно выбираться в зависимости от типа физического уровня, используемого в беспроводной локальной сети.

Как и в методе CSMA/CD, в данном методе количество неудачных попыток передачи одного кадра ограничено, но стандарт 802.11 не устанавливает точного значения этого верхнего предела. Когда верхний предел в N попыток достигнут, кадр отбрасывается, а счётчик последовательных коллизий устанавливается в нуль. Этот счётчик также устанавливается в нуль, если кадр после некоторого количества неудачных попыток все же передаётся успешно.

В беспроводных сетях возможна ситуация, когда два устройства (A и B) удалены и не слышат друг друга, однако оба попадают в зону охвата третьего устройства C. Это так называемая *проблема скрытого терминала*. Если оба устройства A и B начнут передачу, то они принципиально не смогут обнаружить конфликтную ситуацию и определить, почему пакеты не проходят.

В режиме доступа DCF применяются меры для устранения эффекта скрытого терминала. Для этого станция, которая хочет захватить среду и в соответствии с описанным алгоритмом начинает передачу кадра в определённом слоте, вместо кадра данных сначала посылает станции назначения короткий служебный кадр RTS (Request To Send — запрос на передачу). На этот запрос станция назначения должна ответить служебным кадром CTS (Clear To Send — свободна для передачи), после чего станция-отправитель посылает кадр данных. Кадр CTS должен оповестить о захвате среды те станции, которые находятся вне зоны сигнала станции-отправителя, но в зоне досягаемости станции-получателя, т.е. являются скрытыми терминалами для станции-отправителя.

Максимальная длина кадра данных 802.11 равна 2346 байт, длина RTS-кадра — 20 байт, CTS-кадра — 14 байт. Так как RTS- и CTS-кадры гораздо короче, чем кадр данных, потери данных в результате коллизии RTS- или CTS-кадров гораздо меньше, чем при коллизии кадров данных. Процедура обмена RTS- и CTS-кадрами не обязательна. От неё можно отказаться при небольшой нагрузке сети, поскольку в такой ситуации коллизии случаются редко.

Централизованный режим доступа PCF. Если в сети имеется точка доступа, то может применяться централизованный режим доступа PCF, обеспечивающий приоритетное обслуживание трафика. В этом случае точка доступа играет роль арбитра среды.

Режим доступа PCF в сетях 802.11 сосуществует с режимом DCF. Оба режима координируются с помощью трёх типов межкадровых интервалов.

После освобождения среды каждая станция отсчитывает время простоя среды, сравнивая его с тремя значениями:

– короткий межкадровый интервал (Short IFS, SIFS);
– межкадровый интервал режима PCF (PIFS);
– межкадровый интервал режима DCF (DIFS).

Захват среды с помощью распределённой процедуры DCF возможен только в том случае, когда среда свободна в течение времени, равного или большего, чем DIFS. То есть в качестве IFS в режиме DCF нужно использовать интервал DIFS — самый длительный период из трёх возможных, что даёт этому режиму самый низкий приоритет.

Межкадровый интервал SIFS имеет наименьшее значение, он служит для первоочередного захвата среды ответными CTS-кадрами или квитанциями, которые продолжают или завершают уже начавшуюся передачу кадра.

Значение межкадрового интервала PIFS больше, чем SIFS, но меньше, чем DIFS. Промежутком времени между завершением PIFS и DIFS пользуется арбитр среды. В этом промежутке он может передать специальный кадр, который говорит всем станциям, что начинается контролируемый период. Получив этот кадр, станции, которые хотели бы воспользоваться алгоритмом DCF для захвата среды, уже не могут этого сделать, они должны дожидаться окончания контролируемого периода. Его длительность объявляется в специальном кадре, но этот период может закончиться и раньше, если у станций нет чувствительного к задержкам трафика. В этом случае арбитр передаёт служебный кадр, после которого по истечении интервала DIFS начинает работать режим DCF.

На управляемом интервале реализуется централизованный метод доступа PCF. Арбитр выполняет процедуру опроса, чтобы по очереди предоставить каждой такой станции право на использование среды, направляя ей специальный кадр. Станция, получив такой кадр, может ответить другим кадром, который подтверждает приём специального кадра и одновременно передаёт данные (либо по адресу арбитра для транзитной передачи, либо непосредственно станции).

Для того чтобы какая-то доля среды всегда доставалась асинхронному трафику, длительность контролируемого периода ограничена. После его окончания арбитр передаёт соответствующий кадр, и начинается неконтролируемый период.

Каждая станция может работать в режиме PCF, для этого она должна подписаться на данную услугу при присоединении к сети.

Типы кадров MAC

Контрольные кадры. Способствуют надёжной доставке информационных кадров. Существует шесть подтипов контрольных кадров:

- *Опрос после выхода из экономичного режима (PS-опрос).* Данный кадр передаётся любой станцией станции, включающей точку доступа. В кадре запрашивается передача кадра, прибывшего, когда станция находилась в режиме энергосбережения, и в данный момент размещённого в буфере точки доступа.
- *Запрос передачи (RTS).* Данный кадр является первым из четвёрки, используемой для обеспечения надёжной передачи данных. Станция, пославшая это сообщение, предупреждает адресата и остальные станции, способные принять данное сообщение, о своей попытке передать адресату информационный кадр.
- *Готов к передаче (CTS).* Второй кадр четырёхкадровой схемы. Передаётся станцией-адресатом станции-источнику и предоставляет право отправки информационного кадра.
- *Подтверждение (ACK).* Подтверждение успешного приёма предыдущих данных, кадра управления или кадра PS-опроса.
- *Без состязания (CF-конец).* Объявляет конец периода без состязания; часть стратегии использования распределённого режима доступа.
- *CF-конец + CF-подтверждение.* Подтверждает кадр CF-конец. Данный кадр завершает период без состязания и освобождает станции от ограничений, связанных с этим периодом.

Информационные кадры. Существует восемь подтипов информационных кадров, собранных в две группы. Первые четыре подтипа определяют кадры, переносящие данные высших уровней от исходной станции к станции-адресату.

- *Данные.* Просто информационный кадр. Может использоваться как в период состязания, так и в период без состязания.
- *Данные + CF-подтверждение.* Может передаваться только в период без состязания. Помимо данных, в этом кадре имеется подтверждение полученной ранее информации.
- *Данные + CF-опрос.* Используется точечным координатором для доставки данных к мобильной станции и для запроса у мобильной станции информационного кадра, который находится в её буфере.
- *Данные + CF-подтверждение + CF-опрос.* Объединяет в одном кадре функции двух описанных выше кадров.

Остальные четыре подтипа информационных кадров фактически не переносят данные пользователя.

- Информационный кадр *нулевая функция* не переносит ни данных, ни запросов, ни подтверждений. Он используется только для передачи точке доступа бита управления питанием в поле управления кадром, указывая, что станция перешла в режим работы с пониженным энергопотреблением.
- Оставшиеся три кадра (*CF-подтверждение*, *CF-опрос*, *CF-подтверждение + CF-опрос*) имеют те же функции, что и описанные выше

подтипы кадров (*данные + CF-подтверждение*, *данные + CF-опрос*, *данные + CF-подтверждение + CF-опрос*), но не несут пользовательских данных.

Кадры управления. Кадры управления используются для управления связью станций и точек доступа.

- *Запрос ассоциации.* Посылается станцией к точке доступа с целью запроса ассоциации с данной сетью с *базовым набором услуг (Basic Service Set, BSS)*. Кадр включает информацию о возможностях, например, будет ли использоваться шифрование или способна ли станция отвечать при опросе.
- *Ответ на запрос ассоциации.* Возвращается точкой доступа и указывает, что запрос ассоциации принят.
- *Запрос повторной ассоциации.* Посылается станцией при переходе между BSS, когда требуется установить ассоциацию с точкой доступа в новом BSS. Использование повторной ассоциации, а не просто ассоциации, позволяет новой точке доступа договариваться со старой о передаче информационных кадров по новому адресу.
- *Ответ на запрос повторной ассоциации.* Возвращается точкой доступа и указывает, что запрос повторной ассоциации принят.
- *Пробный запрос.* Используется станцией для получения информации от другой станции или точки доступа. Кадр используется для локализации BSS стандарта IEEE 802.11.
- *Ответ на пробный запрос.* Отклик на пробный запрос.
- *Сигнальный кадр.* Передаётся периодически, позволяет мобильным станциям локализовать и идентифицировать BSS.
- *Объявление наличия трафика.* Посылается мобильной станцией с целью уведомления других (которые могут находиться в режиме пониженного энергопотребления), что в буфере данной станции имеются кадры, адресованные другим.
- *Разрыв ассоциации.* Используется станцией для аннулирования ассоциации.
- *Аутентификация.* Для аутентификации станций используются множественные кадры.
- *Отмена аутентификации.* Передаётся для прекращения безопасного соединения.

Подстандарты

Стандарт IEEE 802.11b благодаря высокой скорости передачи данных, практически эквивалентной пропускной способности обычных проводных локальных сетей Ethernet, а также ориентации на диапазон 2,4 ГГц, завоевал наибольшую популярность у производителей оборудования для беспроводных сетей.

Поскольку оборудование, работающее на максимальной скорости 11 Мбит/с, имеет меньший радиус действия, чем на более низких скоростях, то стандартом 802.11b предусмотрено автоматическое снижение скорости при ухудшении качества сигнала.

Стандарт IEEE 802.11a имеет наибольшую ширину полосы пропускания из семейства стандартов 802.11 при скорости передачи данных до 54 Мбит/с.

В отличие от базового стандарта, ориентированного на область частот 2,4 ГГц, спецификациями 802.11a предусмотрена работа в диапазоне 5 ГГц. В качестве метода модуляции сигнала выбрано ортогональное частотное мультиплексирование (OFDM).

К недостаткам 802.11a относятся более высокая потребляемая мощность радиопередатчиков для частот 5 ГГц, а также меньший радиус действия.

Стандарт IEEE 802.11g является логическим развитием 802.11b и предполагает передачу данных в том же частотном диапазоне. Кроме того, стандарт 802.11g полностью совместим с 802.11b, т.е. любое устройство 802.11g должно поддерживать работу с устройствами 802.11b. Максимальная скорость передачи в стандарте 802.11g составляет 54 Мбит/с.

При разработке стандарта 802.11g рассматривались две отчасти конкурирующие технологии: *метод ортогонального частотного разделения OFDM* и *метод двоичного пакетного свёрточного кодирования PBCC*, опционально реализованный в стандарте 802.11b. В результате стандарт 802.11g содержит компромиссное решение: в качестве базовых применяются технологии OFDM и CCK, а опционально предусмотрено использование технологии PBCC.

Набор стандартов 802.11 определяет целый ряд технологий реализации *физического уровня (Physical Layer Protocol, PHY)*:

- уровень PHY стандарта 802.11 со скачкообразной перестройкой частоты (FHSS) в диапазоне 2,4 ГГц;
- уровень PHY стандарта 802.11 с расширением спектра методом прямой последовательности (DSSS) в диапазоне 2,4 ГГц;
- уровень PHY стандарта 802.11b с комплементарным кодированием в диапазоне 2,4 ГГц;
- уровень PHY стандарта 802.11a с ортогональным частотным мультиплексированием (OFDM) в диапазоне 5 ГГц;
- расширенный физический уровень (Extended Rate Physical Layer, ERP) стандарта 802.11g в диапазоне 2,4 ГГц.

Каждый из физических уровней стандарта 802.11 имеет два подуровня:

- *процедуру определения состояния физического уровня* (Physical Layer Convergence Procedure, PLCP);
- *подуровень физического уровня, зависящий от среды передачи* (Physical Medium Dependent, PMD).

Подуровень PLCP, по существу, является уровнем обеспечения взаимодействия, на котором осуществляется перемещение *элементов данных протокола MAC (MAC Protocol Data Units, MPDU)* между MAC-станциями с использованием подуровня PMD, на котором реализуется тот или иной метод передачи и приёма данных через беспроводную среду. Подуровни PLCP и PMD отличаются для разных вариантов стандарта 802.11.

Одна из особенностей, лежащих в основе современных передатчиков, благодаря которой данные можно передавать с высокой скоростью, — это предположение о том, что данные, которые предлагаются для передачи, поступают, с точки зрения передатчика, случайным образом. Без этого предположения многие преимущества, получаемые за счёт применения остальных составляющих физического уровня, остались бы нереализованными.

Скрэмблирование (перестановка элементов) — метод, посредством которого принимаемые данные делаются более похожими на случайные; достигается это путём перестановки битов последовательности таким образом, чтобы превратить её из структурированной в похожую на случайную. Дескрэмблер приёмника затем выполняет обратное преобразование этой случайной последовательности с целью получения исходной структурированной последовательности. Большинство способов скрэмблирования относится к числу самосинхронизирующихся; это означает, что дескрэмблер способен самостоятельно синхронизироваться со скрэмблером.

Исходный стандарт 802.11 определяет три метода передачи на физическом уровне:

- передачу в диапазоне инфракрасных волн;
- технологию расширения спектра путём скачкообразной перестройки частоты (FHSS) в диапазоне 2,4 ГГц;
- технологию широкополосной модуляции с расширением спектра методом прямой последовательности (DSSS) в диапазоне 2,4 ГГц.

Передача в диапазоне инфракрасных волн. Средой передачи являются инфракрасные волны диапазона 850 нм, которые генерируются либо полупроводниковым лазерным диодом, либо светодиодом (LED). Область покрытия LAN ограничивается зоной прямой видимости. Стандарт предусматривает три варианта распространения излучения: ненаправленную антенну, отражение от потолка и фокусное направленное излучение. В первом случае узкий луч рассеивается с помощью системы линз. Фокусное направленное излучение предназначено для организации двухточечной связи.

Беспроводные локальные сети со скачкообразной перестройкой частоты (FHSS). Беспроводные локальные сети FHSS поддерживают скорости передачи 1 и 2 Мбит/с. Устройства FHSS делят

предназначенную для их работы полосу частот от 2,402 до 2,480 ГГц на 79 неперекрывающихся каналов. Ширина каждого из 79 каналов составляет 1 МГц, поэтому беспроводные локальные сети FHSS используют относительно высокую скорость передачи символов (1 МГц) и намного меньшую скорость перестройки с канала на канал.

Последовательность перестройки частоты должна иметь следующие параметры: частота перескоков не менее 2,5 раз в секунду как минимум между шестью (6 МГц) каналами. Чтобы минимизировать число коллизий между перекрывающимися зонами покрытия, возможные последовательности перескоков должны быть разбиты на три набора последовательностей, длина которых для Северной Америки и большей части Европы составляет 26.

По сути, схема скачкообразной перестройки частоты обеспечивает медленный переход с одного возможного канала на другой таким образом, что после каждого скачка покрывается полоса частот, равная как минимум 6 МГц, благодаря чему в многосотовых сетях минимизируется возможность возникновения коллизий.

В спецификации стандарта 802.11 оговорено использование и другого физического уровня — на основе технологии широкополосной модуляции с расширением спектра методом прямой последовательности (DSSS). Как было указано в стандарте 802.11 разработки 1997 г., технология DSSS поддерживает скорости передачи 1 и 2 Мбит/с.

IEEE 802.11b. Накладные расходы в этом стандарте выше, чем в проводной сети Ethernet. Поэтому крайне важно обеспечить высокую скорость передачи данных в канале. Повысить пропускную способность канала с заданной шириной полосы частот можно, разрабатывая и применяя новые методы модуляции. По этому пути пошла группа разработчиков IEEE 802.11b.

Изначально стандарт IEEE 802.11 предусматривал работу в режиме DSSS с использованием так называемой Баркеровской последовательности (Barker) длиной 11 бит: B1 = (0b10110111000). Каждый информационный бит замещается своим произведением по модулю 2 (XOR) с данной последовательностью. В результате бит заменяется последовательностью 11 чипов. Далее сигнал кодируется посредством дифференциальной двух- или четырёхпозиционной фазовой модуляции (DBPSK или DQPSK, один или два чипа на символ соответственно). При частоте модуляции несущей 11 МГц общая скорость составляет, в зависимости от типа модуляции, 1 и 2 Мбит/с.

Стандарт IEEE 802.11b дополнительно предусматривает скорости передачи 11 и 5,5 Мбит/с. Для этого используется так называемая *CCK-модуляция (Complementary Code Keying — кодирование комплементарным кодом)*.

Стандарт IEEE 802.11a появился практически одновременно с IEEE 802.11b, в сентябре 1999 г. Эта спецификация была ориентирована

на работу в диапазоне 5 ГГц и основана на принципиально ином, чем описано выше, механизме кодирования данных — на частотном мультиплексировании посредством ортогональных несущих (OFDM).

Стандарт 802.11a определяет характеристики оборудования, применяемого в офисных или городских условиях, когда распространение сигнала происходит по многолучевым каналам из-за множества отражений.

В IEEE 802.11a каждый кадр передаётся посредством 52 ортогональных несущих, каждая с шириной полосы порядка 300 кГц (20 МГц/64). Ширина одного канала — 20 МГц. Несущие модулируют посредством BPSK, QPSK, а также 16- и 64-позиционной квадратурной амплитудной модуляции (QAM). В совокупности с различными скоростями кодирования (1/2 и 3/4, для 64-QAM — 2/3 и 3/4) образуется набор скоростей передачи 6, 9, 12, 18, 24, 36, 48 и 54 Мбит/с. Из 52 несущих 48 предназначены для передачи информационных символов, остальные 4 — служебные.

IEEE 802.11g. Стандарт IEEE 802.11g по сути представляет собой перенесение схемы модуляции OFDM, зарекомендовавшей себя в 802.11a, из диапазона 5 ГГц в область 2,4 ГГц при сохранении функциональности устройств стандарта 802.11b. Это возможно, поскольку в стандартах 802.11 ширина одного канала в диапазонах 2,4 и 5 ГГц схожа — 22 МГц.

Одним из основных требований к спецификации 802.11g была обратная совместимость с устройствами 802.11b. В качестве основного способа модуляции принята схема *CCK (Complementary Code Keying)*, а в качестве дополнительной возможности допускается модуляция PBSS.

Разработчики 802.11g предусмотрели CCK-модуляцию для скоростей вплоть до 11 Мбит/с и OFDM для более высоких скоростей. Но сети стандарта 802.11 при работе используют принцип CSMA/CA — множественный доступ к каналу связи с контролем несущей и предотвращением коллизий. Ни одно устройство 802.11 не должно начинать передачу, пока не убедится, что эфир в его диапазоне свободен от других устройств. Если в зоне слышимости окажутся устройства 802.11b и 802.11g, причём обмен будет происходить между устройствами 802.11g посредством OFDM, то оборудование 802.11b просто не поймёт, что другие устройства сети ведут передачу, и попытается начать трансляцию. Чтобы не допустить подобной ситуации, предусмотрена возможность работы в смешанном режиме — CCK-OFDM.

Одна из основных проблем стандарта — как обеспечить бесконфликтную работу смешанных сетей 802.11b/g. Основной принцип работы в сетях 802.11 — «слушать, прежде чем вещать». Но устройства 802.11b не способны услышать устройства 802.11g в OFDM-режиме. Ситуация аналогична проблеме скрытых станций: два устройства уда-

лены настолько, что не слышат друг друга и пытаются обратиться к третьему, которое находится в зоне слышимости обоих. Для предотвращения конфликтов в подобной ситуации в 802.11 введён защитный механизм, предусматривающий перед началом информационного обмена передачу короткого кадра *запрос на передачу* (RTS) и получение кадра подтверждения *можно передавать* (CTS). Механизм RTS/CTS применим и к смешанным сетям 802.11b/g. Естественно, эти кадры должны транслироваться в режиме CCK, который обязаны понимать все устройства. Однако защитный механизм существенно снижает пропускную способность сети.

4.8.3. Стек протоколов IEEE 802.16 (WiMAX)

WiMAX (Worldwide Interoperability for Microwave Access) — стандарт беспроводной связи IEEE 802.16 (рабочая группа создана в 1999 г.).

При зазработке ставились следующие задачи:

- обеспечение доступа к услугам информационных и коммуникационных технологий для небольших поселений, удалённых регионов, изолированных объектов;
- обеспечение доступа к услугам информационных и коммуникационных технологий более половины населения планеты.

Преимущества технологии:

- стандарт объединяет технологии как уровня оператора связи, так и технологии «последней мили»;
- беспроводные технологии более гибки и, как следствие, проще в развёртывании, так как по мере необходимости могут масштабироваться;
- простота установки как фактор уменьшения затрат на развёртывание сетей в развивающихся странах, малонаселенных или удалённых районах;
- на данный момент большинство беспроводных технологий широкополосной передачи данных требуют наличия прямой видимости между объектами сети; WiMAX благодаря использованию технологии OFDM создаёт зоны покрытия в условиях отсутствия прямой видимости от клиентского оборудования до базовой станции (диаметр соты порядка нескольких километров);
- изначально содержит протокол IP, что позволяет легко и прозрачно интегрировать её в локальные сети;
- подходит для фиксированных, перемещаемых и подвижных объектов сетей на единой инфраструктуре.

Характеристики:

- пропускная способность до 135 Мбит/с при полосе несущей 28 МГц;
- доступ к среде адаптивный, динамический;
- управление сетью централизованное.

Принципы работы WiMAX

Соединение между базовой станцией и клиентским приёмником производится в СВЧ-диапазоне 2–11 ГГц. Такое соединение в идеальных условиях позволяет передавать данные со скоростью до 20 Мбит/с и не требует, чтобы станция находилась на расстоянии прямой видимости от пользователя. Этот режим работы базовой станции WiMAX близок широко распространённому стандарту IEEE 802.11, что допускает совместимость уже выпущенных клиентских устройств и WiMAX.

Между соседними базовыми станциями устанавливается постоянное соединение с использованием сверхвысокой частоты 10–66 ГГц радиосвязи *прямой видимости*. Такое соединение в идеальных условиях позволяет передавать данные со скоростью до 120 Мбит/с.

Как минимум одна из базовых станций может быть постоянно связана с сетью провайдера через широкополосное скоростное соединение. Даже при небольшом количестве точек система способна корректно распределить нагрузку за счёт сотовой топологии.

На базе сотового принципа разрабатываются также пути построения оптимальной сети, огибающей крупные объекты, когда серия последовательных станций передаёт данные по эстафетному принципу. По структуре сети стандарта IEEE 802.16 очень похожи на традиционные сети мобильной связи: здесь тоже имеются базовые станции, которые действуют в радиусе до 50 км, при этом их также необязательно устанавливать на вышках. Для них вполне подходят крыши домов, требуется лишь соблюдение условия прямой видимости между станциями. Для соединения базовой станции с пользователем необходимо наличие абонентского оборудования. Далее сигнал может поступать по стандартному Ethernet-кабелю как непосредственно на конкретный компьютер, так и на точку доступа стандарта IEEE 802.11 или в локальную проводную сеть стандарта Ethernet, что позволяет сохранить существующую инфраструктуру районных или офисных локальных сетей при переходе с кабельного доступа на WiMAX.

Режимы работы

Стандарт 802.16e-2005[1] вобрал в себя все ранее выходившие версии и на данный момент предоставляет следующие режимы:

- стационарный доступ (Fixed WiMAX) (рабочая группа IEEE 802.16d, стандарт IEEE 802.16-2004);
- сеансовый доступ (Nomadic[2] WiMAX);
- доступ в режиме перемещения (Portable WiMAX);

[1] Документ IEEE 802.16e-2005 сам по себе является не стандартом, а дополнением к стандарту IEEE 802.16-2004.

[2] Буквально: кочующий.

– мобильный доступ (Mobile WiMAX) (дополнение IEEE 802.16e-2005).

Fixed WiMAX. Фиксированный доступ представляет собой альтернативу широкополосным проводным технологиям (xDSL, T1). Стандарт использует диапазон частот 10–66 ГГц. Этот частотный диапазон из-за сильного затухания коротких волн требует прямой видимости между передатчиком и приёмником сигнала, но позволяет избежать одной из главных проблем радиосвязи — многолучевого распространения сигнала. При этом ширина каналов связи в этом частотном диапазоне довольно велика (типичное значение — 25 или 28 МГц), что позволяет достигать скоростей передачи до 120 Мбит/с. Фиксированный режим включался в версию стандарта IEEE 802.16d-2004.

Nomadic WiMAX. Сеансовый доступ добавил понятие сессий к уже существующему Fixed WiMAX. Наличие сессий позволяет свободно перемещать клиентское оборудование между сессиями и восстанавливать соединение уже с помощью других вышек WiMAX, нежели те, что использовались во время предыдущей сессии. Такой режим разработан в основном для портативных устройств. Введение сессий позволяет также уменьшить расход энергии клиентского устройства.

Portable WiMAX. Для режима Portable WiMAX добавлена возможность автоматического переключения клиента от одной базовой станции WiMAX к другой без потери соединения. Однако для данного режима все ещё ограничена скорость передвижения клиентского оборудования — 40 км/ч.

Mobile WiMAX. Этот режим был разработан в стандарте 802.16e-2005 и позволил увеличить скорость перемещения клиентского оборудования до 120 км/ч. Основные достижения этого режима:
– устойчивость к многолучевому распространению сигнала и собственным помехам;
– масштабируемая пропускная способность канала;
– технология *Time Division Duplex (TDD)*, которая позволяет эффективно обрабатывать асимметричный трафик и упрощает управление сложными системами антенн за счёт эстафетной передачи сессии между каналами;
– технология *Hybrid-Automatic Repeat Request (H-ARQ)*, которая позволяет сохранять устойчивое соединение при резкой смене направления движения клиентского оборудования;
– распределение выделяемых частот и использование субканалов при высокой загрузке позволяет оптимизировать передачу данных с учётом силы сигнала клиентского оборудования;

- управление энергосбережением позволяет оптимизировать затраты энергии на поддержание связи портативных устройств в режиме ожидания или простоя;
- технология *Network-Optimized Hard Handoff (HHO)*, которая позволяет до 50 мс и менее сократить время на переключение клиента между каналами;
- технология *Multicast and Broadcast Service (MBS)*, которая объединяет функции DVB-H, MediaFLO и 3GPP E-UTRA для:
  - достижения высокой скорости передачи данных с использованием одночастотной сети;
  - гибкого распределения радиочастот;
  - низкого потребления энергии портативными устройствами;
  - быстрого переключения между каналами;
- технология *Smart Antenna*, поддерживающая субканалы и эстафетную передачу сессии между каналами, что позволяет использовать сложные системы антенн, включая формирование диаграммы направленности, пространственно-временное маркирование, пространственное мультиплексирование;
- технология *Fractional Frequency Reuse*, которая позволяет контролировать наложение/пересечение каналов для повторного использования частот с минимальными потерями;
- размер фрейма в 5 мс обеспечивает компромисс между надёжностью передачи данных за счёт использования малых пакетов и накладными расходами посредством увеличения числа пакетов (и, как следствие, заголовков).

4.8.4. Технология Bluetooth

Bluetooth представляет собой беспроводную технологию, обеспечивающую беспроводную передачу данных на небольших расстояниях между различными устройствами (например, мобильными персональными компьютерами, мобильными телефонами и другими устройствами) в режиме реального времени. При этом возможна передача как цифровых данных, так и звуковых сигналов.

Работа над концепцией системы Bluetooth началась в 1994 г. шведской компанией Ericsson. В феврале 1998 г. по инициативе пяти ведущих зарубежных компаний — Ericsson, IBM, Intel, Nokia и Toshiba — была организована специальная группа (Special Interest Group, SIG) [35], в задачи которой входило продвижение этой технологии. В мае того же года последовало объявление об учреждении концерна Bluetooth. В конце 1999 г. появились первые спецификации на соответствующее оборудование, ставшие впоследствии стандартом де-факто. На спецификациях Bluetooth v. 1.x базируется стандарт IEEE 802.15.1[1], утверждённый в 2002 г.

[1]IEEE 802.15 Working Group for WPAN. URL: `http://www.ieee802.org/15/`.

Устройства версий 1.0 и 1.0B имели плохую совместимость между продуктами различных производителей. Основным недостатком была невозможность реализовать анонимность на протокольном уровне.

В Bluetooth 1.1 было исправлено множество ошибок, найденных в 1.0B, добавлена поддержка для нешифрованных каналов, *индикация уровня мощности принимаемого сигнала (Received Signal Strength Indicator, RSSI)*.

В версии 1.2 (2003 г.) была добавлена *технология адаптивной перестройки рабочей частоты (Adaptive Frequency-Hopping Spread Spectrum, AFH)*, что улучшило сопротивляемость к электромагнитной интерференции (помехам) путём использования разнесённых частот в последовательности перестройки. Также увеличилась скорость передачи и добавилась технология *eSCO (Extended Synchronous Connections)*, которая улучшила качество передачи голоса путём повторения повреждённых пакетов. В *HCI (Host Controller Interface)* добавилась поддержка трёхпроводного интерфейса *UART (Universal Asynchronous Receiver/Transmitter)*.

Основные отличия Bluetooth 1.2: ускоренное установление соединения, адаптивная схема переключения каналов (от 20 до 79), усовершенствованные алгоритмы передачи данных.

Версия Bluetooth 2.0+EDR (2004 г.) состоит из двух частей, которые могут поддерживаться аппаратурой независимо: обновлённая версия спецификации Bluetooth (без принципиальных отличий от версии 1.2) и *расширенный набор скоростей передачи данных (Enhanced Data Rate, EDR)*. В режиме EDR применяется дифференциальная фазовая модуляция, увеличивающая базовую скорость передачи с 1 до 3 Мбит/с.

Стандарт Bluetooth 2.0+EDR полностью совместим с Bluetooth 1.0 и 1.2; скорость передачи в пикосети не ограничивается скоростью самого медленного.

В 2007 г. появилась обновлённая версия Bluetooth 2.1 (полное название Bluetooth Core Specification Version 2.1 + EDR). Эта версия полностью совместима с версией 2.0. В ней удалось снизить энергопотребление, а также усовершенствовать алгоритм связи.

Основные параметры радиоинтерфейса Bluetooth

– Диапазон частот: 2,4–2,4835 ГГц — промышленный, научный и медицинский диапазон частот (Industrial, Scientific and Medical band, ISM band)[1].

[1]ISM band определяет полосы частот (918, 2450 и 5800 МГц, 22,5 ГГц) для работы промышленной, научной и медицинской радиослужб.

- Число несущих частот: 23–79 с разносом 1 МГц (16/32 в одной пикосети).
- Метод доступа: скачкообразная перестройка частоты и *дуплексная передача с временны́м разделением каналов (Frequency-hopping spread spectrum / Time-Division Duplex, FHSS/TDD)* (1600 скачков в секунду).
- Метод модуляции: *частотная модуляция с гауссовским сглаживанием (Gaussian Frequency Shift Keying, GFSK)* — двухуровневая схема кодирования сигнала, в которой логическому 0 и 1 соответствуют две разные частоты, коэффициент сглаживания формы входных импульсов $h = 0,35$.
- Скорость передачи по радиоканалу: 1 Мбит/с.
- Полоса пропускания: 220 кГц (по уровню 3 дБ), 1 МГц (по уровню 20 дБ).
- Мощность передатчика: 100 мВт (для связи до 100 м; 20 дБм), 2 мВт (до 10 м; 4 дБм) и 1 мВт (10 см; 0 дБм).

Принцип работы

Абонентские устройства Bluetooth объединяются в группы (пикосети), совместно использующие один радиоканал. В состав каждой пикосети входят один ведущий приёмопередатчик (с опорным генератором, который синхронизирует внутренний трафик сети) и до семи ведомых (синхронизируемых). Ведомое устройство вычисляет разность между частотами собственного и ведущего генераторов, и в процессе вхождения в синхронизм эта погрешность учитывается, что обеспечивает точное соответствие излучаемой частоты данного и ведущего устройств.

Вид псевдослучайной последовательности однозначно идентифицирует ведущий приёмопередатчик, а её фаза (псевдослучайный сдвиг) является адресным признаком ведомого устройства. Период повторения последовательности, определяющей закон перестройки частоты, достаточно большой (свыше 23 ч.). В каждой пикосети используется своя псевдослучайная последовательность, что позволяет множеству пикосетей одновременно работать по одному и тому же каналу связи, не создавая взаимных помех.

Синхронное и асинхронное соединения

Bluetooth имеет возможность организовывать как синхронное, так и асинхронное соединение.

Синхронное соединение (Synchronous Connection Oriented, SCO) возможно только в режиме точка–точка и применяется для передачи

информации, чувствительной к задержкам (например, голоса). Основное (ведущее) устройство поддерживает до трёх синхронных соединений, подчинённое — до трёх синхронных соединений с одним основным устройством или до двух — с разными основными устройствами. При синхронном соединении основное устройство резервирует временны́е сегменты, следующие через так называемые SCO-интервалы. Даже если пакет принят с ошибкой, повторно при синхронном соединении он не передаётся.

Асинхронное соединение (Asynchronous Connection Less, ACL) возможно между основным и всеми активными подчинёнными устройствами в пикосети (режим точка–многоточка). Основное и подчинённое устройства могут поддерживать только одно асинхронное соединение. Подчинённое устройство отправляет пакет основному, только если в предыдущем временно́м интервале на его адрес пришёл пакет от основного устройства. Асинхронное соединение позволяет повторно передавать пакеты, принятые с ошибками.

Таким образом, Bluetooth может поддерживать один асинхронный канал данных (со скоростью до 723,2 Кбит/с в прямом и 57,6 Кбит/с в обратном направлениях), до трёх синхронных (с постоянной скоростью 64 Кбит/с в каждом направлении) голосовых каналов или канал с одновременной асинхронной передачей данных и синхронной передачей голоса (со скоростью до 433,9 Кбит/с в каждом направлении).

Структура кадра

Стандартный кадр Bluetooth содержит *код доступа (Access Code)* (длина 72 бита), *заголовок* (длина 54 бита) и *информационное поле* (длина не более 2745 бит) (рис. 4.21).

Рис. 4.21. Стандартный кадр Bluetooth

Код доступа идентифицирует пакеты, принадлежащие одной пикосети, а также используется для синхронизации и процедуры запросов; состоит из трёх полей:
- преамбулы (Preamble) (4 бита),
- кода синхронизации (Sync Word) (64 бита),
- концевика (Trailer) (4 бита контрольной суммы).

Заголовок (рис. 4.22) содержит информацию для управления связью и состоит из шести полей:

– поле *Адрес (AM_ADDR)* (3 бита) — содержит MAC-адрес узла назначения;
– поле *Тип пакета (TYPE)* (4 бита) — указывает код одного из 12 типов данных (ACL, SCO, опрос или пустой кадр), метод коррекции ошибок и число временны́х интервалов, из которых состоит кадр;
– поле *Поток (FLOW)* (1 бит) — осуществляет управление потоком данных, показывает готовность устройства к приёму;
– поле *Признак повторной передачи (ARQ)* (1 бит) — определяет корректность приёма;
– поле *SEQN* (1 бит) — служит для определения последовательности пакетов;
– поле *Контроль ошибок в заголовке (Header Error Check, HEC)* (8 бит) — контрольная сумма.

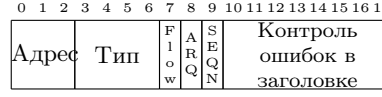


Рис. 4.22. Заголовок Bluetooth

Информационное поле имеет три сегмента:
– поле *Заголовок полезной информации* (8 бит) определяет логический канал, управление потоком в логических каналах, а также имеет указатель длины полезной информации;
– поле *Тело полезной информации* (0–2721 бит) включает пользовательскую информацию; длина этого сегмента указана в поле длины заголовка полезной информации;
– поле *Циклический избыточный код (Cyclic Redundancy Check, CRC)* (16 бит) служит для контроля целостности передаваемых данных.

Существует шесть типов пакетов Bluetooth.

Протоколы Bluetooth

Иерархия протоколов Bluetooth не соответствует моделям ISO/OSI, TCP/IP и IEEE 802. В спецификации определено 5 уровней: физический, базовый, управления каналом, сетевой и уровень приложений.

На *физическом уровне* определены параметры радиоинтерфейса Bluetooth.

Базовый уровень (baseband) и *уровень управления связью (Link Control Layer)* обеспечивают физическую радиочастотную связь между устройствами Bluetooth, образующими пикосеть.

На базовом уровне определено 13 типов пакетов. Пакеты ID, NULL, POLL, FHS , DM1 определены как для синхронных, так и для асинхронных соединений. Пакеты DH1, AUX1, DM3, DH3, DM5 и DH5

определены только для асинхронного соединения. Форматы пакетов HV1, HV2, HV3 и DV определены только для синхронного соединения. Кроме того, на данном уровне определены пять логических каналов: *LC (Control Channel)* и *LM (Link Manager)* используются на канальном уровне, а *UA (User Asynchronous)*, *UI (User Isosynchronous)* и *US (User Synchronous)* служат для асинхронной, изосинхронной и синхронной транспортировки пользовательских данных.

Протокол управления связью (Link Manager Protocol, LMP) отвечает за установление подключений между устройствами Bluetooth. Также сюда относятся и вопросы безопасности, такие как идентификация и шифрование, связанные с генерированием ключей шифрования и подключения, а также с обменом ключами и их проверкой. Кроме того, протокол контролирует режимы питания и исполнительные циклы устройств Bluetooth, а также состояние подключения того или иного устройства к пикосети.

Протокол управления логическим подключением и адаптацией (Logical Link Control and Adaptation Protocol, L2CAP) адаптирует протоколы верхнего уровня над Baseband. L2CAP является базовым протоколом передачи данных для Bluetooth. L2CAP работает только с ACL-соединениями. Многие протоколы и службы более высокого уровня используют L2CAP как транспортный протокол.

Протокол обнаружения услуг (Service Discovery Protocol, SDP) использует L2CAP в качестве транспортного протокола, что позволяет запросить информацию о самом устройстве, его услугах и характеристиках этих услуг, а после этого может быть установлено соединение между двумя или несколькими устройствами Bluetooth.

Протокол замены кабеля (RFCOMM) также использует L2CAP в качестве транспортного протокола. Протокол RFCOMM эмулирует соединение PPP (point-to-point) по последовательному порту, обеспечивает транспортировку при выполнении услуг верхнего уровня.

Двоичный протокол управления телефонией (Telephony Control Protocol Specification Binary, TCS Binary) является биториентированным протоколом и определяет контроль сигнализации вызова для установления речевого вызова или вызова данных между устройствами Bluetooth. Кроме того, он определяет процедуры управления мобильностью при манипулировании с группами TCS-устройств Bluetooth.

Профили Bluetooth

Профили Bluetooth представляют собой общие механизмы (протоколы и функции), через которые доступные устройства Bluetooth взаимодействуют с другими устройствами. Профили определяют области возможного применения устройства Bluetooth. Если устройства от различных производителей соответствуют одному профилю, определённому в спецификации Bluetooth, они смогут взаимодействовать

друг с другом.

- *Профиль общего доступа (Generic Access Profile, GAP)*
  Профиль GAP отвечает за поддержание связи между устройствами, выявление других доступных профилей, а также за безопасность соединений. Этот профиль должен быть включён во все устройства Bluetooth. В него входят функции, необходимые для работы всех основных протоколов Bluetooth.
- *Профиль последовательного порта (Serial Port Profile, SPP)*
  Профиль SPP позволяет устройствам Bluetooth эмулировать последовательный порт при помощи протокола RFCOMM. Профиль SPP определяет, каким образом два доступных устройства Bluetooth будут осуществлять обмен данными посредством эмуляции интерфейса RS-232 или интерфейса USB.
- *Профиль приложения обнаружения услуг (Service Discovery Application Profile, SDAP)*
  Профиль SDAP описывает, каким образом приложение должно использовать *протокол обнаружения услуг (Service Discovery Protocol, SDP)*. Профиль SDAP необходим для того, чтобы любое приложение имело возможность узнать, какие услуги (сервисы) Bluetooth является доступными на любом устройстве Bluetooth, с которым оно соединено.
- *Общий профиль обмена объектами (Generic Object Exchange Profile, GOEP)*
  Профиль GOEP используется для непосредственного (без использования IP) обмена объектами между двумя устройствами. Объект может иметь любой тип, например, изображение, документ, визитная карточка и т.д. Профиль определяет устройству одну из двух ролей: сервер, который определяет место, куда объект был помещён, и клиент, который инициализирует механизм передачи.
- *Профиль дозвона по сети (Dial-Up Networking Profile, DUN)*
  DUN обеспечивает стандартный доступ к сети Интернет и другому сервису модемной связи по беспроводной технологии Bluetooth.
- *Профиль факсимильной связи (Fax Profile, FAX)*
  Профиль FAX определяет, каким образом устройство, имеющее шлюз факсимильного аппарата, может использоваться в качестве оконечного устройства. Профиль FAX предназначен для обеспечения интерфейса между мобильным телефоном (или телефоном стационарной сети) и персональным компьютером с установленным программным обеспечением, поддерживающим факс.
- *Профиль гарнитуры (Headset Profile, HSP)*
  Профиль HSP определяет способ, посредством которого Bluetooth обеспечивает беспроводное соединение устройства с гарнитурой, оснащённой динамиками и, возможно, микрофоном.
- *Профиль доступа к локальной сети (LAN Access Profile, LAP)*
  Профиль LAP предназначен для создания IP-сетей и позволяет

создавать небольшие беспроводные сети Intranet, объединяющие ПК или смарт-телефоны. Он также используется точками доступа для связи с кабельными сетями, будь то локальные сети или Internet.

– *Профиль передачи файлов (File Transfer Profile, FTP)*
Профиль FTP определяет, каким образом файлы на устройстве сервера могут быть просмотрены устройством клиента. Если местонахождение файла определено клиентом, то файл может быть перемещён от сервера к клиенту или помещён клиентом на сервер, используя профиль GOEP.
– *Профиль помещения объектов в стек (Open Push Profile, OPP)*
Профиль OPP управляет обменом электронными визитками в формате vCard (расширение файлов *.vcf). Эти визитки содержат ту же информацию, что и традиционные, но при этом они могут быть автоматически занесены в личную информационную систему (PIM) или в базу данных.
– *Профиль синхронизации (Synchronization Profile, SYNC)*
Профиль SYNC используется вместе с GOEP, чтобы обеспечить синхронизацию календаря и адресной информации (элементы управления персональной информации — PIM) между доступными Bluetooth-устройствами. Основное применение этого профиля — обмен данными между персональным цифровым секретарём (PDA) и компьютером.
– *Профиль беспроводной телефонной связи (Cordless Telephony Profile, CTP)*
Профиль CTP определяет, каким образом беспроводной телефон может быть использован в технологии Bluetooth. Этот профиль может использоваться или для беспроводного телефона, или для мобильного телефона, который функционирует как беспроводной телефон вблизи от базовой станции.
– *Профиль внутренней связи (Intercom Profile, ICP)*
Этот профиль обеспечивает двустороннюю голосовую связь между устройствами Bluetooth. Он рассчитан на прямое взаимодействие двух устройств, расположенных в зоне взаимной досягаемости. Технология была разработана таким образом, чтобы, с одной стороны, не создавать ненужных помех для других пользователей, а с другой — быть невосприимчивым к радиосигналам других технологий, работающих на этих же частотах.

Дополнительные профили Bluetooth для устройств печати:

– *Профиль замены кабеля твёрдой копии (Hard Copy Cable Replacement Profile, HCRP)*
Профиль HCRP обеспечивает беспроводной вариант связи в качестве замены кабельного соединения между устройством и принтером.
– *Основной профиль принтера (Basic Printing Profile, BPP)*
Профиль BPP обеспечивает механизм формирования заданий выво-

да на печать текстов, сообщений электронной почты, изображений, визиток типа vCards и других объектов. Отличие этого профиля от HCRP заключается в том, что BPP не требует наличия специфических драйверов для каждого конкретного принтера.

Дополнительные профили Bluetooth для аудио- и видеоаппаратуры:

– *Общий профиль распространения аудио и видео (General Audio/Video Distribution Profile, GAVDP)*
Профиль GAVDP является основой для профилей A2DP и VDP, применяемых в системах распределения видео- и аудиопотоков, использующих беспроводную технологию Bluetooth.
– *Расширенный профиль распространения аудио (Advanced Audio Distribution Profile, A2DP)*
Профиль A2DP описывает, каким образом качественный стереозвук проходит от источника до приёмника.
– *Профиль распространения видео (Video Distribution Profile, VDP)*
Профиль VDP определяет, каким образом доступное Bluetooth-устройство обеспечивает передачу потоков видеоинформации, используя Bluetooth-технологии.
– *Профиль дистанционного управления аудио- и видеоаппаратурой (Audio/ Video Remote Control Profile, AVRCP)*
Профиль AVRCP обеспечивает стандартный интерфейс для управления высококачественной аудио- и видеоаппаратурой. Использование этого профиля позволяет единственному пульту дистанционного управления осуществлять управление всей аудио- и видеоаппаратурой, которая находится в окрестности. Профиль AVRCP даёт возможность управлять характеристиками мультимедиа потоков, например, регулировкой громкости, пуском, приостановкой и остановкой плейера, а также выполнять другие подобные операции дистанционного управления.

Основную конфигурацию дополняют другие профили Bluetooth:

– *Основной профиль изображения (Basic Imaging Profile, BIP)*
Профиль BIP обеспечивает механизм дистанционного управления устройствами записи, передачи и отображения изображений (например, управление затвором цифровой фотокамеры с помощью мобильного телефона). Добавляется в основную конфигурацию профилей под управление профиля GOEP.
– *Профиль Hands-Free (Hands-Free Profile, HFP)*
Профиль HFP описывает, каким образом устройство-шлюз может использоваться для размещения и получения вызовов устройства hands-free. Типичный пример — применение мобильного телефона в качестве устройства-шлюза. Профиль HFP позволяет также использовать ресурсы мультимедиа персонального компьютера в качестве аппаратуры громкой связи мобильного телефона. Добавляется в основную конфигурацию профилей под управление профиля SPP.

4.9. Краткие итоги раздела

1. Канальный уровень (Data Link) обеспечивает обмен данными через общую локальную среду. Он разделен на два подуровня (LLC и MAC).
2. Подуровень LLC реализует связь с протоколами сетевого уровня.
3. Подуровень MAC определяет особенности доступа к физической среде при использовании различных технологий локальных сетей.
4. Каждой технологии MAC-уровня соответствует несколько вариантов (спецификаций) протоколов физического уровня, которые определяют скорость передачи, вид среды.
5. На MAC подуровне современных сетей используется ряд технологий: Ethernet, Fast Ethernet, Gigabit Ethernet и 10Gigabit Ethernet.
6. В локальных сетях адресация узлов производится на основе MAC-адресов, содержащих 48 двоичных разрядов. MAC-адреса представлены в шестнадцатеричной системе.
7. В сетях технологии Ethernet, построенных на основе логической топологии «общая шина», разделяемая среда передачи данных является общей для всех пользователей. При этом реализуется метод множественного доступа к среде с контролем несущей и обнаружением коллизий (CSMA/CD).
8. Для предотвращения коллизий современные локальные сети строятся на базе коммутаторов, которые делят сеть на сегменты коллизий.
9. Продвижение кадров с входного интерфейса коммутатора на выходной происходит на основании записей в адресной таблице коммутации.
10. Различные режимы коммутации позволяют изменять производительность коммутатора.
11. Протокол для предотвращения петель в коммутируемых сетях STP используется в сетях с избыточными путями.
12. В качестве среды передачи в локальных сетях передачи данных используют коаксиальный кабель, неэкранированную UTP и экранированную STP витую пару (симметричный кабель), оптоволоконный кабель, беспроводные радиоканалы.
13. Для соединения устройств между собой используются прямой, кроссовый и консольный кабели.
14. Волоконно-оптические кабели характеризуются отсутствием перекрестных помех и электромагнитных помех от внешних источников. Это позволяет передавать сигналы на большее расстояние по сравнению с симметричным медным кабелем.
15. Передача данных по оптическому волокну производится на длинах волн 850, 1310 или 1550 нм.
16. В технологии Fast Ethernet сохранился принцип использования общей разделяемой среды. Основными спецификациями технологии Fast Ethernet являются 100Base-TX и 100Base-FX.

17. Начало кадра протокола Fast Ethernet отделяется от символов свободной среды Idle парой символов J и K (11000 и 10001) кода 4B/5B, а конец – символом T.
18. Стандарт 802.3z технологии Gigabit Ethernet определил две спецификации 1000Base-SX и 1000Base-LX. Для использования уже имеющихся симметричных кабелей UTP категории 5 был разработан стандарт 802.3ab.
19. Технология 10-Gigabit Ethernet (10GbE) описывается стандартом IEEE 802.3ae, который определяет полнодуплексную передачу данных со скоростью 10 Гбит/с по волоконно-оптическому кабелю.

4.10. Вопросы по разделу

1. Укажите функции и услуги канального уровня модели ISO/OSI.
2. Приведите классификацию методов доступа к среде. Опишите схемы работы основных методов доступа к среде.
3. Как осуществляется управление логической передачей данных на канальном уровне?
4. Укажите различия в форматах кадра Ethernet.
5. В чем отличия технологий Fast Ethernet и Gigabit Ethernet от Ethernet?
6. Объясните схему функционирования сетей с маркерным доступом. В чем сходство и отличия архитектур и принципов работы сетей Token Bus, Token Ring, FDDI?
7. Какие элементы составляют архитектуру сети на базе технологии 100VG-AnyLAN и какие функции они выполняют? Укажите отличия методов доступа в технологиях 100VG-AnyLAN и Fast Ethernet.
8. Объясните принцип работы технологий доступа с виртуальными каналами.
9. Какие элементы составляют архитектуру сети на базе технологии X.25 и какие функции они выполняют?
10. Какие элементы составляют архитектуру сети на базе технологии Frame Relay и какие функции они выполняют?
11. Опишите основные технологии региональных сетей. В чем их принципиальные отличия?

4.11. Примеры заданий

Задание 4.1 — Определите тип MAC-адреса Ethernet A1:AA:C7:F8:D0:05.

Ответ (Задание 4.1) — Последний бит первого байта равен 1. Следовательно, это групповой адрес. Предпоследний бит первого байта равен 0. Следовательно, это глобально администрируемый адрес.

4.12. Задания для самостоятельной работы

Задание 4.2 — Перечислите спецификации технологий Ethernet, Fast Ethernet. Приведите их основные характеристики.

Задание 4.3 — Изобразите формат кадра MAC-подуровня технологии Ethernet.

Задание 4.4 — Объясните, почему задается минимальная длина поля данных.

Задание 4.5 — Изобразите схему локальной сети на коммутаторе с пятью конечными узлами, укажите номера портов и MAC-адреса узлов. Создайте таблицу коммутации для случая, когда все узлы активно обмениваются данными.

Глава 5. Сетевой уровень

Глава посвящена протоколам, отвечающим за адресацию и маршрутизацию сетевого уровня модели взаимодействия открытых систем. Подробно рассмотрены структуры заголовков протоколов IPv4, IPv6, структура и принципы организации адресного пространства. Кратко рассмотрены протоколы ICMP, ARP, RARP. Достаточно подробно рассмотрены типы, принципы и протоколы маршрутизации (RIP, OSPF, BGP). Дополнительно даны сведения о технологии коммутации по меткам (MPLS).

В основу главы легли материалы из источников [1; 2], а также стандарты по IPv4[1], IPv6[2] и материалы по Click и MPLS[3].

[1] Internet Protocol, RFC 791. URL: `http://www.faqs.org/rfcs/rfc791.html`; *Braden R.*, *Postel J.* Requirements for Internet gateways, RFC 1009. URL: `http://www.faqs.org/rfcs/rfc1009.html`; *Hinden R.* Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR), RFC 1517. URL: `http://www.faqs.org/rfcs/rfc1517.html`; *Rekhter Y.*, *Li T.* An Architecture for IP Address Allocation with CIDR, RFC 1518. URL: `http://www.faqs.org/rfcs/rfc1518.html`; Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, RFC 1519 / V. Fuller, T. Li, J. Yu, K. Varadhan. URL: `http://www.faqs.org/rfcs/rfc1519.html`; *Rekhter Y.*, *Topolcic C.* Exchanging Routing Information Across Provider Boundaries in the CIDR Environment, RFC 1520. URL: `http://www.faqs.org/rfcs/rfc1520.html`.

[2] *Deering S.*, *Hinden R.* Internet Protocol, Version 6 (IPv6) Specification, RFC 1883. URL: `http://www.faqs.org/rfcs/rfc1883.html`; *Deering S.*, *Hinden R.* Internet Protocol, Version 6 (IPv6) Specification, RFC 2460. URL: `http://www.faqs.org/rfcs/rfc2460.html`; *Hinden R.*, *Deering S.* Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513. URL: `http://www.faqs.org/rfcs/rfc3513.html`; *Blanchet M.* A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block, RFC 3531. URL: `http://www.faqs.org/rfcs/rfc3531.html`; *Hinden R.*, *Deering S.*, *Nordmark E.* IPv6 Global Unicast Address Format, RFC 3587. URL: `http://www.faqs.org/rfcs/rfc3587.html`; OSI NSAPs and IPv6, RFC 1888 / J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd. URL: `http://www.faqs.org/rfcs/rfc1888.html`; *Narten T.*, *Draves R.* Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC 3041. URL: `http://www.faqs.org/rfcs/rfc3041.html`; Адресация протокола IPv6. 2005. URL: `http://msdn.microsoft.com/ru-ru/library/cc775951(v=ws.10).aspx`.

[3] The Click Modular Router Project. URL: `http://www.read.cs.ucla.edu/click/`; *Rosen E.*, *Viswanathan A.*, *Callon R.* Multiprotocol Label Switching Architecture, RFC 3031. URL: `http://www.ietf.org/rfc/rfc3031.txt`; MPLS Label Stack Encoding, RFC 3032 / E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta. URL: `http://www.ietf.org/rfc/rfc3032.txt`; LDP Specification, RFC 3036 / L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas. URL: `http://www.ietf.org/rfc/rfc3036.txt`; *Юшков Т.* Архитектура MPLS. URL:

В результате освоения данной темы с учётом выполнения заданий и лабораторного практикума (разделы 9.3 и 9.4) студент должен:

знать:

– структуры заголовков протоколов межсетевого уровня стека TCP/IP;
– принципы организации адресного пространства IPv4 и IPv6;
– принципы коммутации по меткам (технология MPLS);
– принципы построения сетей передачи данных и настройки сетевого оборудования;

уметь:

– планировать адресное пространство IPv4;
– определять тип адреса IPv6;

владеть:

– способностью планировать структуру сети передачи данных;
– способностью настраивать коммутационное и маршрутизирующее оборудование;
– способностью анализировать данные заголовков сетевых протоколов.

5.1. Протокол IPv4

Межсетевой протокол IP (Internet Protocol или IPv4 (IP Version 4)) описан в RFC 791[1] и является базовым протоколом стека TCP/IP.

Первоначальное назначение этого протокола — передача пакетов в гетерогенных (составных) сетях со сложной топологией. Он не гарантирует доставку данных до узла назначения, т.е. является датаграммным протоколом.

5.1.1. Формат пакета IP

Пакет протокола IP имеет заголовок длиной до 20 байт и поле данных. Структура заголовка IP-пакета приведена на рис. 5.1.

В поле *Версия (Version)* (длина 4 бита) указана версия протокола IP.

В поле *Длина заголовка (IHL)* (длина 4 бита) указывается длина заголовка в 32-битных словах[2]. Обычная длина заголовка — 20 байт (т.е. 5 32-битовых слов), но в заголовок может быть включена дополнительная информация, расположенная в поле *Опции*.

`http://www.mpls-exp.ru/mplsarchitecture.html`.

[1] Internet Protocol, RFC 791. URL: `http://www.faqs.org/rfcs/rfc791.html`.

[2] В четырёх битах можно максимально закодировать число 15.

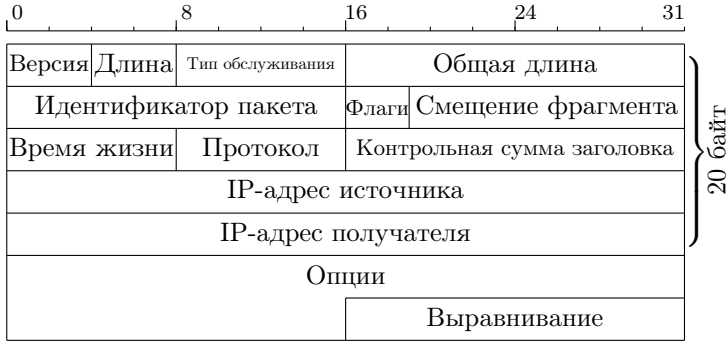


Рис. 5.1. Формат заголовка пакета IPv4

Таким образом, максимальная длина заголовка может составлять 60 байт (т.е. 15 32-битовых слов)[1].

В поле *Тип обслуживания (Type of Service, TOS)* (длина 8 бит) имеется несколько подполей (рис. 5.2). В подполе *приоритет (Precedence)* (длина 3 бита) указывается значение приоритета передаваемых данных от самого низкого — 0 (обычный пакет) до самого высокого — 7 (пакет с управляющей информацией). В первую очередь обрабатываются пакеты более высокого приоритета.

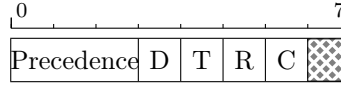


Рис. 5.2. Поле *Тип обслуживания* заголовка IP

После поля приоритета пакета идут четыре бита, которые задают тип обслуживания. Только один из четырёх бит может иметь значение 1. Первый из этих четырёх бит указывает, что должна быть *малая задержка (Low Delay, D)* во время передачи, второй — *высокая пропускная способность (High Throughput, T)*, третий — *высокая надёжность (High Reliability, R)*, четвёртый — *низкая стоимость (Low*

[1]Может показаться, что для заголовка IP достаточно 60 байт в любых случаях. Однако это не так. Например, при использовании поля опции *Запись маршрута (IP Record Route)* в заголовке нельзя записать более девяти пройденных на маршруте точек.

Cost, C). Последний бит подполя был добавлен уже после появления RFC 791[1]. Смысл значений типа обслуживания абстрактен[2].

Последний бит поля не используется.

В поле *Общая длина (Total Length)* (длина 16 бит) задаётся общий размер пакета в байтах, при этом учитывается длина заголовка и поля данных. Наибольшее значение длины пакета ограничено разрядностью этого поля и и не может превышать 65535 байт, но обычно такие большие пакеты не пересылаются.

В поле *Идентификатор пакета (Identification)* (длина 16 бит) указывается номер каждого пересылаемого пакета IP. Этот номер увеличивается каждый раз на 1 при очередной посылке пакета, за исключением пересылки фрагментированных IP-пакетов, в которых значение в данном поле одинаковое для всех фрагментов. Для идентификации фрагментов используются поля *Флаги* и *Смещение фрагмента*.

Поле *Флаги (Flags)* (длина 3 бита) (рис. 5.3) бит *Не фрагментировать (Do not Fragment, DF)* запрещает маршрутизатору разбивать пакет на части, а бит *Есть ещё фрагменты (More Fragments, MF)* указывает, что пакет не последний среди фрагментов. Первый бит зарезервирован.

Рис. 5.3. Поле *Флаги* заголовка IP

В поле *Смещение фрагмента (Fragment Offset)* (длина 13 бит) задаётся, на сколько байт поле данных смещено от начала исходного фрагментирующегося пакета. Смещение должно быть кратным 8 байтам.

В поле *Время жизни (Time to Live, TTL)* (длина 8 бит) задан максимальный срок нахождения пакета в сети. Уменьшение значения этого поля происходит или каждую секунду, или при прохождении пакета через промежуточный маршрутизатор. При достижении значения этого поля 0, происходит отбрасывание этого пакета маршрутизатором, а источнику пакета отправляется уведомление об истечении его

[1] Internet Protocol, RFC 791. URL: `http://www.faqs.org/rfcs/rfc791.html`.

[2] В RFC 791 отмечено: «Тип обслуживания — это абстрактный и обобщённый набор параметров, характеризующий услуги, которые предоставляются сетями, составляющими объединённую сеть. Значения из поля *Тип обслуживания* должны использовать шлюзы при выборе параметров реальной пересылки информации в данной сети, сети следующего участка на пути пакета или следующего шлюза при маршрутизации сетевой датаграммы».

времени жизни в виде пакета ICMP, основное назначение которого — предотвратить зацикливание маршрута пакета между маршрутизаторами.

Пакет ICMP также направляется источнику данных в случае, когда на узле-получателе принята только часть фрагментов пакета IP, но время жизни истекло. В этом случае отправка уведомляющего сообщения происходит в период ожидания начала сборки пакета на узле-получателе.

Поле *Протокол верхнего уровня (Protocol)* (8 бит) содержит идентификатор протокола верхнего уровня, которому принадлежит информация, содержащаяся в поле данных пакета. Значения идентификаторов для различных протоколов приведены в RFC «Assigned Numbers» (табл. 5.1).

Таблица 5.1

Идентификаторы наиболее распространённых протоколов

Значение	Название	Описание
1	ICMP	Internet Control Message Protocol — протокол межсетевых управляющих сообщений
2	IGMP	Internet Group Menagement Protocol — протокол управления межсетевыми группами
3	GGP	Gateway-to-Gateway Protocol — межшлюзовый протокол
4	IP	Инкапсуляция IP в IP
6	TCP	Transmission Control Protocol — протокол управления передачей
8	EGP	Exterior Gateway Protocol — протокол внешнего шлюза
17	UDP	User Datagram Protocol — протокол пользовательских датаграмм
88	IGRP	Interior Gateway Routing Protocol — внутренний протокол маршрутизации
89	OSPF	Open Shortest Path First

Значение контрольной суммы рассчитывается без учёта поля данных, т.е. только по заголовку IP-пакета, и размещается в поле *Контрольная сумма (Header Checksum)* (длина 2 байта). Значение этого

поля пересчитывается при прохождении промежуточных узлов на маршруте движения IP-пакета, так как значение некоторых полей его заголовка может меняться при движении пакета по сети. В процессе вычисления контрольной суммы значение самого поля *Контрольная сумма* обнуляется.

Адреса отправителя и получателя размещаются в соответствующих полях *IP-адрес источника (Source IP Address)* и *IP-адрес назначения (Destination IP Address)* (длина по 32 бита).

Поле *Опции (Option)* — необязательное, но может использоваться во время отладки. Поле имеет несколько подполей с предопределёнными типами. К число подполей может меняться, поэтому после поля *Опции* может быть поле *Выравнивание (Padding)*, предназначенное для выравнивания нулями заголовка по 32-битной границе.

5.1.2. Схема адресации протокола IPv4

Для организации взаимосвязи устройств в IP-сети используется IP-адресация сетевых интерфейсов. Устройство с несколькими сетевыми интерфейсами должно иметь соответствующее число IP-адресов.

Схема адресации протокола IPv4 приведена в документах RFC 990[1], RFC 997[2].

Длина IP-адреса — 32 бита. Для удобства принято записывать IP-адрес в *десятично-точечной нотации*: каждый из четырёх байт (октет) IP-адреса записывается десятичным числом в диапазоне от 0 до 255; октеты разделяются точками (например, 10.130.10.2).

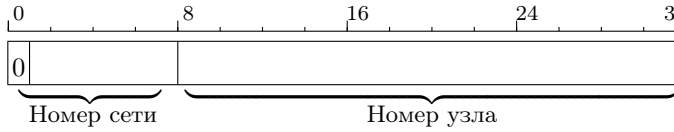
Классы адресов

Определено 5 классов IP-адресов: A, B, C, D, E. Адреса классов A, B и C имеют две логические части: *номер сети* и *номер узла* (рис. 5.4, табл. 5.2).

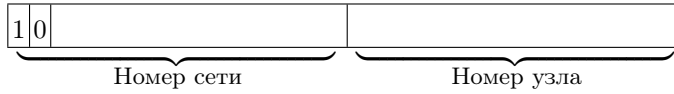
Старший бит адресов класса A имеет значение 0 (рис. 5.4a). Сетевой префикс имеет длину 8 бит. Под номер узла выделено 24 бита. Два номера сети имеют специальное значение, см. п. 5.1.2, поэтому очевидно, что в классе A может быть 126 сетей ($2^7 - 2$). При это в каждой сети этого класса максимально возможное число узлов — 16777214 ($2^{24} - 2$, также см. п. 5.1.2). Адресный пул класса A может состоять максимум из 2^{31} уникальных адресов (в протоколе IPv4 всего может быть 2^{32} адресов).

[1] *Reynolds J.*, *Postel J.* Assigned numbers, RFC 990. URL: `http://www.faqs.org/rfcs/rfc990.html`.

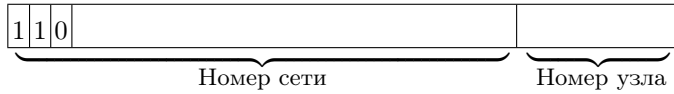
[2] *Reynolds J.*, *Postel J.* Internet numbers, RFC 997. URL: `http://www.faqs.org/rfcs/rfc997.html`.



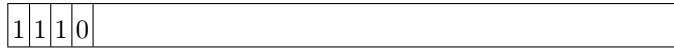
(а) Класс A



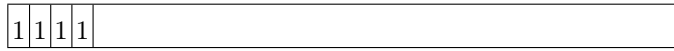
(б) Класс B



(в) Класс C



(г) Класс D



(д) Класс E

Рис. 5.4. Классы сетей IPv4

Таблица 5.2

Классы IP-адресов

Класс адреса	Формат записи[a]	Старшие биты	Границы адресов сети	Количество битов в адресе сети/хоста	Максимальное количество сетей	Максимальное количество хостов	Назначение
A	N.H.H.H	0	от 1.0.0.0 до 126.0.0.0	8/24	$2^7 - 2$	$2^{24} - 2$	Большие организации
B	N.N.H.H	1 0	от 128.0.0.0 до 191.255.0.0	16/16	2^{14}	$2^{16} - 2$	Организации среднего размера
C	N.N.N.H	1 1 0	от 192.0.0.0 до 223.255.255.0	24/8	2^{21}	$2^8 - 2$	Малые организации
D	—	1 1 1 0	от 224.0.0.0 до 239.255.255.255	—	—	—	Групповое вещание
E	—	1 1 1 1	от 240.0.0.0 до 247.255.255.255	—	—	—	Экспериментальные

[a] [a]Обозначения: N — часть адреса, относящаяся к номеру сети, H — часть адреса, относящаяся к номеру узла

Адресное пространство класса A содержит 50% от адресного пространства протокола IPv4. Назначение адресов класса A — сети с большим числом узлов. На данный момент все адреса класса A распределены.

Первые два бита адресов класса B установлены соответственно в 1 и 0 (рис. 5.4б). Сетевой префикс имеет длину 16 бит, поле номера узла — 16 бит. В сети класса B может быть 16384 (2^{14}) сетей, каждая из которых может содержать до 65534 узлов ($2^{16}-2$). Адресный пул сетей класса B может состоять максимально из 2^{30} уникальных адресов, т.е. 25% всего адресного пространства. Адреса класса B применяются в сетях среднего размера.

Три старших бита адресов класса C имеют значение соответственно 1, 1 и 0 (рис. 5.4в). Длина префикса — 24 бита, номер узла занимает 8 бит. В классе C может иметь максимально 2097152 (2^{21}) сетей, каждая из которых может поддерживать до 254 уникальных адресов узлов (2^8-2). Пул адресов сетей класса C содержит 2^{29} уникальных адреса, т.е. 12,5% всего адресного пространства протокола IPv4. Назначение адресов класса C — сети с небольшим числом узлов.

Структура адресов классов D и E другая.

Старшие биты адресов класса D имеют значение соответственно 1, 1, 1 и 0 (рис. 5.4г). Адреса класса D являются идентификаторами группы при передаче данных по схеме «один-ко-многим» (*групповое вещание (Multicasting)*). Узлы могут относится к разным сетям, но принадлежать к одной группе вещания.

Четыре старших бита адресов класса E имеют значение 1, 1, 1 и 1 (рис. 5.4д). Класс E зарезервирован для экспериментов.

Служебные IP-адреса

Часть IP-адресов зарезервирована (табл. 5.3) для служебных целей.

Интерпретируются служебные адреса следующим образом:

1. При направлении данных устройством самому себе для идентификации этого устройства используется адрес, все биты которого установлены в 0 (например, 0.0.0.0).
2. Для идентификации узлов одной и той же сети используются адреса, все биты поля номера сети которых установлены в 0 (например, 0.0.0.2).
3. Для рассылки данных всем узлам сети отправителя используется *ограниченный широковещательный адрес (Limited Broadcast)*, в котором все биты установлены в 1 (255.255.255.255).
4. Для идентификации сети используются адреса, в которых все биты поля номера узла установлены в 0, а в поле адреса сети стоит адрес сети (например, 10.130.128.0/32).

5. Для рассылки *широковещательных сообщений (Broadcast)* используются адреса, в которых все биты в поле номера узла установлены в 1, а в поле адреса сети стоит адрес сети (например, 10.130.128.255/32). Из этих двух пунктов видно, что в любой сети два значения номера узла зарезервированы для служебной надобности.
6. Для взаимодействия процессов внутри узла используется *возвратный (Loopback)* адрес, первый октет которого имеет значение 127 (например, 127.0.0.0/8).

Таблица 5.3

Служебные IP-адреса

Поле сети	Поле узла	Интерпретация
Все биты равны 0	Все биты равны 0	Данное устройство
Все биты равны 0	Номер узла	Устройство в данной IP-сети
Все биты равны 1	Все биты равны 1	Все устройства в данной IP-сети (ограниченный широковещательный адрес (Limited Broadcast))
Номер сети	Все биты равны 0	Данная IP-сеть
Номер сети	Все биты равны 1	Все устройства в указанной IP-сети (широковещательный адрес (Broadcast))
127		Возвратный адрес (Loopback)

Другие особые адреса:
- В случае имеющейся в операционной системе настройки получения IP-адреса по протоколу DHCP, но отсутствия ответа от сервера DHCP, узлу присваивается адрес типа 169.254.0.0/16.
- Адреса в частных сетях: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.

Подсети

Стандартная классовая схема разбиения адресного пространства приводит к ряду проблем, таких как:
- резкое увеличение числа таблиц маршрутизации в Интернете;
- неэффективное использование адресного пространства.

С целью решения возникших проблем был добавлен дополнительный уровень иерархии структуры IP-адреса — номер подсети (рис. 5.5).

При таком подходе адресация сети организации снаружи (до шлюза) осуществляется по номеру сети, а внутри организации возможно

разбиение сети на подсети, что не видно снаружи. В этом случае адресное пространство используется более оптимально, что решает в некоторой степени вторую проблему. Изменения адресного пространства внутри организации не оказывают влияние на маршрутизацию в Интернете, что в некоторой степени решает первую проблему.

Рис. 5.5. Двухуровневая и трёхуровневая иерархии IP-адресов

Маска подсети

Совокупность номера сети и номера подсети представляет собой *Расширенный сетевой префикс*, для выделения которого из адреса применяется операция XOR (побитное сложение по модулю два) IP-адреса и *маски подсети*.

Маска подсети (Subnet Mask) — это сетевой адрес, в котором биты номера сети и подсети установлены в единицу, а остальные биты — в ноль.

IP-адрес полностью определяется IP-адресом и маской подсети. Число подсетей определяется степенью двойки — 2^n, где n — длина поля номера подсети.

Для упрощения записи применяют так называемую *CIDR-нотацию*):

`IP-адрес/длина_ расширенного_сетевого_префикса`.

Например, адрес 10.128.0.0 с маской 255.255.0.0 будет в данной нотации выглядеть как 10.128.0.0/16 (очевидно, что 16 — это число единиц в маске подсети).

Далее приведены маски подсетей для стандартных классов сетей (в десятично-точечной нотации):

- 255.0.0.0 — маска для сети класса A; длина расширенного сетевого префикса — 8;
- 255.255.0.0 — маска для сети класса B; длина расширенного сетевого префикса — 16;
- 255.255.255.0 — маска для сети класса C; длина расширенного сетевого префикса — 24.

Маска подсети переменной длины

RFC 1009[1] позволил, используя *маски подсетей переменной длины (Variable Length Subnet Mask, VLSM)*, разбивать сети на подсети разного размера.

Выделение подсетей производится рекурсивно: исходная сеть разбивается на несколько подсетей, затем некоторые из них также делятся на подсети и т.д. Число доступных подсетей определяется по формуле 2^m, где m — количество бит, выделяемых для идентификации сети, а количество узлов в сети вычисляется по формуле $2^n - 2$, где n — количество бит, выделяемых для идентификации узла.

На рис. 5.6 проиллюстрировано разбиение сети класса C.

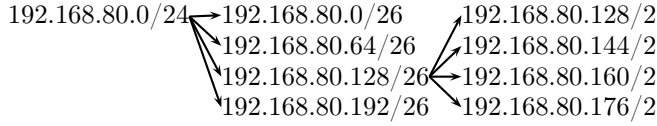


Рис. 5.6. Разбиение сети на подсети

Рассмотрим подробнее разбиение сети на несколько подсетей.

Пример (Разбиение сети класса A на 5 подсетей).

Изначально пусть задана сеть 32.0.0.0/8. В двоичной форме адрес сети запишется следующим образом

`00100000 00000000 00000000 00000000`.

Расширенный префикс имеет длину 8 бит. Сеть имеет маску 255.0.0.0 (или в двоичной форме `11111111 00000000 00000000 00000000`) и broadcast-адрес 32.255.255.255/8 (или в двоичной форме `00100000 11111111 11111111 11111111`) и её можно разбить на $2^8 = 256$ подсетей.

Сначала выделим 4 подсети. Для этого добавим к идентификатору сети ещё 2 бита, чтобы получить следующие 4 комбинации: `00`, `01`, `10`, `11`, которые определят вид второго октета адреса: `00000000` даст 0 в десятичной форме записи, `01000000` — число 64, `10000000` — число 128, `11000000` — число 192.

Длина сетевого префикса составит 10 бит, а маска будет иметь вид 255.192.0.0 (или в двоичной форме `11111111 11000000 00000000 00000000`).

Для получения broadcast-адреса зафиксируем сетевую часть адреса, а биты, относящиеся к номеру хоста, положим равными 1. Тогда сеть 32.64.0.0/10 в двоичной форме запишется следующим образом

[1] *Braden R., Postel J.* Requirements for Internet gateways, RFC 1009. URL: `http://www.faqs.org/rfcs/rfc1009.html`.

00100000 01000000 00000000 00000000.

Её broadcast-адрес в двоичной форме запишется следующим образом

00100000 01111111 11111111 11111111,

а в десятичной форме — 32.127.255.255.

Таким образом, получим 4 подсети.

адрес подсети	broadcast-адрес	маска
32.0.0.0/10	32.63.255.255/10	255.192.0.0
32.64.0.0/10	32.127.255.255/10	255.192.0.0
32.128.0.0/10	32.191.255.255/10	255.192.0.0
32.192.0.0/10	32.255.255.255/10	255.192.0.0

Подсеть 32.64.0.0/10 разделим ещё пополам. Дополнительно выделим 1 бит под идентификатор сети. Получим маску 255.224.0.0 и ещё 2 подсети:

адрес подсети	broadcast-адрес	маска
32.64.0.0/11	32.81.255.255/11	32.224.0.0
32.96.0.0/11	32.127.255.255/11	32.224.0.0

Бесклассовая маршрутизация

Логическим продолжением концепции подсетей является концепция *бесклассовой маршрутизации (Classless InterDomain Routing, CIDR)*, RFC 1517–1520[1]. Основные положения данной технологии:

– Отход от традиционной концепции разделения IP-адресов на классы. Вместо этого для определения границ между номером сети и номером хоста используется расширенный сетевой префикс. Таким образом, возможна организация сетей произвольного размера. При маршрутизации каждая часть маршрутной информации распространяется маршрутизаторами совместно с сетевым префиксом.

[1] *Hinden R.* Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR), RFC 1517. URL: `http://www.faqs.org/rfcs/rfc1517.html`; *Rekhter Y., Li T.* An Architecture for IP Address Allocation with CIDR, RFC 1518. URL: `http://www.faqs.org/rfcs/rfc1518.html`; Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, RFC 1519 / V. Fuller, T. Li, J. Yu, K. Varadhan. URL: `http://www.faqs.org/rfcs/rfc1519.html`; *Rekhter Y., Topolcic C.* Exchanging Routing Information Across Provider Boundaries in the CIDR Environment, RFC 1520. URL: `http://www.faqs.org/rfcs/rfc1520.html`.

– Объединение маршрутов. Рекомендуется выделять адреса иерархически.

5.2. Протокол IPv6

Протокол IPv6 оперирует 128-битными адресами. Помимо решения проблемы с нехваткой адресов в IPv4 этот протокол решает и ряд других задач: позволяет лучше масштабировать сеть, имеет механизмы обеспечения качества обслуживания и информационной безопасности и пр.

Протокол IPv6 имеет следующие отличия от IPv4:
– заголовок IP-пакета имеет меньше полей;
– адресное пространство увеличено с 32 до 128 бит;
– улучшена поддержка иерархической адресации, агрегирования маршрутов и автоматического конфигурирования адресов;
– появилась возможность аутентификации и шифрования на уровне IP-пакетов;
– в качестве идентификаторов добавились метки передаваемых потоков данных.

5.2.1. Формат заголовка пакета IPv6

Так же как и в IPv4 пакет протокола IPv6 имеет заголовок и поле данных. RFC-2460[1] (ранее — RFC-1883[2]) определяет формат заголовка IPv6, длина которого составляет 40 байт. Структура заголовка приведена на рис. 5.7.

Версия протокола IP указывается в поле *Версия (Version)* длиной 4 бита.

Для работы с классами трафика и приоритетами передаваемых пакетов используется поле *Класс трафика (Traffic Class)* длиной 8 бит.

Для идентификации специально обрабатываемого в последствии передаваемого потока данных используется поле *Метка потока (Flow Label)* длиной 20 бит. Под специальной обработкой потока в данном случае понимается наложение ограничений, например, на полосу пропускания или величину задержки при передаче через промежуточные сетевые узлы. Обычно в качестве метки потока выступает некоторое псевдо-случайное число, воспринимаемое сетевыми шлюзами как хеш-ключ, указывающий, как именно должен быть обработан этот поток.

[1] *Deering S., Hinden R.* Internet Protocol, Version 6 (IPv6) Specification, RFC 2460. URL: `http://www.faqs.org/rfcs/rfc2460.html`.

[2] *Deering S., Hinden R.* Internet Protocol, Version 6 (IPv6) Specification, RFC 1883. URL: `http://www.faqs.org/rfcs/rfc1883.html`.

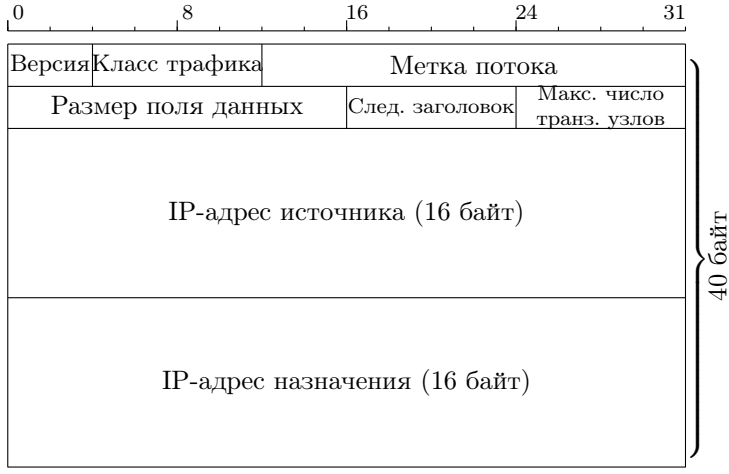


Рис. 5.7. Формат заголовка пакета IPv6 (RFC-2460)

После заголовка пакета следует поле данных, размер которого в байтах задаётся в поле *Размер поля данных (Payload Length)* длиной 16 бит. В случае, если размер поля данных превышает 65535 байт, то значение в поле *Размер поля данных* становится равным нулю, собственно размер поля данных указывается в поле *Сверхдлина (Jumbo Payload)* дополнительного заголовка опций Hop-by-Hop, о котором речь пойдёт позже.

Протокол IPv6 может иметь дополнительные заголовки, предназначенные для выполнения определённых функций. Идентификатор типа дополнительного заголовка указывается в поле *Следующий заголовок (Next Header)* длиной 8 бит.

Аналогом поля времени жизни протокола IPv4 в IPv6 служит поле *Максимальное число транзитных узлов (Hop Limit)* длиной 8 бит. В нём задаётся предельный срок, в течение которого пакет может перемещаться по сети. Значение этого поля уменьшается на 1, когда пакет проходит через промежуточный узел сети (например, шлюз или хост). При достижении нулевого значения этого поля пакет уничтожается.

Адреса отправителя и получателя указываются в соответствующих полях *IP-адрес источника (Source IP Address)* и *IP-адрес назначения (Destination IP Address)* длиной по 128 бит.

5.2.2. Дополнительные заголовки IPv6

После основного заголовка в протоколе IPv6 идут дополнительные заголовки в виде блоков данных, отвечающих за выполнение определённых функций: маршрутизация, фрагментация, аутентификация и др. Каждый из дополнительных заголовков имеет поле, идентифицирующее следующий за ним другой дополнительный заголовок.

Заголовок *опций Hop-by-Hop* используется для передачи узлу дополнительных опций и должен быть обработан каждым узлом на пути следования пакета, включая узлы отправителя и получателя. Этот заголовок имеет идентификатор 0 в основном заголовке IPv6. Его формат приведён на рис. 5.8.

Заголовок состоит из трёх полей. В первом из них, которое называется *Следующий заголовок (Next Header)* и имеет длину 8 бит, размещается идентификатор следующего за ним дополнительного заголовка IPv6 или идентификатор протокола, данные которого включены в поле данных основного заголовка IPv6.

Второе поле называется *Длина дополнительного заголовка (Hdr Ext Len)* и, как видно из названия, содержит размер дополнительного заголовка в 64-битных единицах без учёта первых 64 бит.

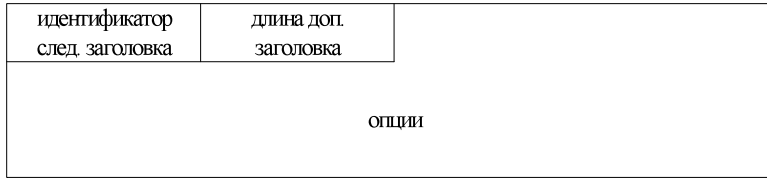


Рис. 5.8. Структура дополнительного заголовка опций Hop-by-Hop

Третье поле *Опции (Options)* имеет переменную длину и содержит ряд параметров для осуществления определённых операций над пакетом. В этом поле указывается тип опции (Option Type, 8 бит), указатель на длину поля данных опции (Opt Data Len, 8 бит) и собственно данные опции (Option Data). Первые два бита подполя типа опции определяют действия узла при невозможности обработки им опций, указанных в заголовке: 00 — пропуск опции и дальнейшая обработка заголовка, 01 — отбрасывание пакета, 10 — отбрасывание пакета и уведомление отправителя сообщением Parameter Problem с помощью протоколы ICMPv6, даже если пакет отправлен на групповой адрес, 11 — отбрасывание пакета и уведомление отправителя сообщением Parameter Problem с помощью протоколы ICMPv6 только в случае, когда пакет отправлен не на групповой адрес.

Одной из возможных опций может быть параметр сверхдлины (Jumbo Payload), указывающий, что содержимое поля данных пакета превышает 65535 байт.

Для определения пути следования пакета IPv6 через промежуточные узлы используется дополнительный заголовок *маршрутизации* (рис. 5.9). Его идентификатор в предыдущем дополнительном заголовке — 43.

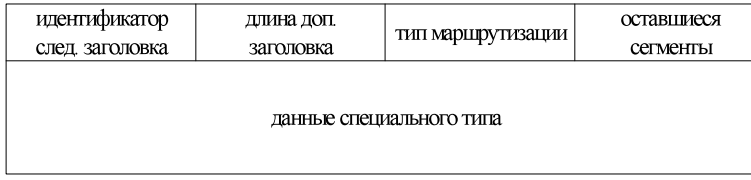


Рис. 5.9. Структура дополнительного заголовка маршрутизации

Заголовок имеет 5 полей. Значения первых двух имеют тот же смысл, что и первые два поля дополнительного заголовка Hop-by-hop Options, т.е. идентифицируют следующий дополнительный заголовок (Next Header, 8 бит) и указывают размер данного заголовка в 64-битных единицах без учёта первого 64-битного блока (Hdr Ext Len, 8 бит).

В поле *Тип маршрутизации (Routing Type)*, имеющим длину 8 бит, указывается идентификатор конкретного варианта маршрутизации, т.е. подтип заголовка.

В поле *Оставшиеся сегменты (Segment Left)* длиной 8 бит указывается число непосещённых узлов маршрута до пункта назначения.

Формат поля *Специальный тип данных (Type-Specific Data)* зависит от содержимого поля *Тип маршрутизации (Routing Type)*.

Идентификатор заголовка *фрагментации* — 44. Данный тип заголовка используется для сборки фрагментированного IPv6 пакета. Пакет фрагментируется, если его размер превышает MTU пути до места назначения. Формат заголовка представлен на рис. 5.10.

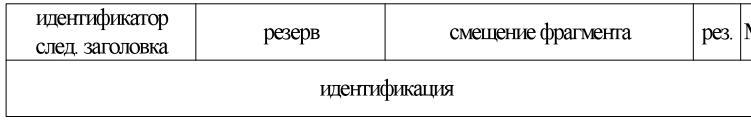


Рис. 5.10. Структура дополнительного заголовка фрагментации

Поле *Смещение фрагмента (Fragment Offset)* указывает смещение фрагмента в 64-битных единицах относительно начала пакета. Идентификатор фрагмента содержится в поле *Идентификация (Identification)*

длиной 32 бита. Флаг *M* указывает на наличие или отсутствие последующих фрагментов.

За функцию обеспечения безопасности в протоколе IPv6 отвечает заголовок *аутентификации*, идентификационный номер которого — 51. Помимо аутентификации гарантируется и целостность передаваемых в IP-пакете данных. Кроме стандартных полей *Следующий заголовок (Next Header)* и *Длина (Length)*, заголовок содержит поля *Индекс параметров безопасности (Security Parameters Index)* длиной 32 бита и *Данные аутентификации (Authentication Data)*.

Поле *Индекс параметров безопасности* содержит псевдослучайное число, которое совместно с IP-адресом получателя и применяемым протоколом безопасности (имеющим характерный тип алгоритма шифрования, параметры шифрования и т.д.) однозначно определяет защищённое виртуальное соединение.

Поле *Данные аутентификации* по сути содержит цифровую подпись, полученную с помощью криптографического ключа с использованием асимметричных методов кодирования (например, RSA), служащую для аутентификации и проверки целостности данных. Размер этого поля должен быть кратен 8 байтам. Для формирования цифровой подписи используется содержимого всех неизменяемых во время перемещения полей IP-пакета.

Заголовок *места назначения* содержит информацию (специальные параметры обработки пакета) для анализа узлом назначения. Его идентификатор — 60. На рис. 5.11 приведён формат заголовка.

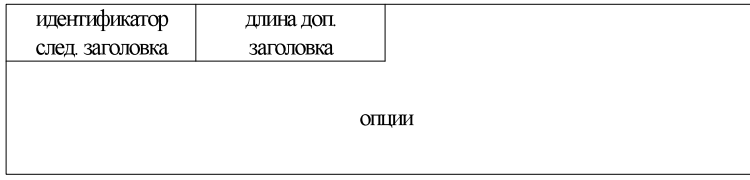


Рис. 5.11. Структура дополнительного заголовка места назначения

Поле *Опции (Options)* данного заголовка имеет такую же структуру, как и аналогичное поле заголовка Hop-by-hop Options.

Наличие заголовка *отсутствия следующего заголовка* показывает отсутствие после него других дополнительных заголовков в IP-пакете. Его идентификатор — 59.

5.2.3. Схема адресации протокола IPv6

Различные схемы адресации протокола IPv6 представлены в документах RFC 3513[1] (ранее RFC-2373, RFC-1884), RFC 3531[2], RFC 3587[3].

Форма записи адреса

Длина IPv6-адреса — 128 бит. При записи адреса используют *двухточечно-шестнадцатеричную* форму: 128 бит разбиты на 8 блоков по 16 бит, которые преобразованы в 4-значные шестнадцатеричные числа, разделённые между собой двоеточием.

IPv6-адрес в двоичной форме:

```
0010000111011010 0000000011010011 0000000000000000
0010111100111011 0000001010101010 0000000011111111
1111111000101000 1001110001011010
```

в двухточечно-шестнадцатеричной форме будет иметь вид:

```
21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A
```

Для сокращения записи в каждом 16-битном блоке убирают начальные нули, но при этом каждый блок должен иметь хотя бы один знак:

```
21DA:D3:0:2F3B:2AA:FF:FE28:9C5A
```

Если в адресе есть длинные последовательности нулей, то их можно сократить до двойного знака двоеточия, например, адрес

```
FE80:0:0:0:2AA:FF:FE9A:4CA2
```

сократится до

```
FE80::2AA:FF:FE9A:4CA2,
```

а укороченная запись адреса `FF02:0:0:0:0:0:0:2` будет иметь вид `FF02::2`.

Очевидно, что укороченную запись длинной сплошной последовательности нулей в адресе можно использовать только один раз, иначе нельзя будет определить число нулей, отображаемых каждым двойным двоеточием.

Понятие префикса

В IPv6 по аналогии с IPv4 имеется возможность разделения пространства адресов по типам и назначению. Для этого используется *префикс формата (Format Prefix, FP)* (рис. 5.12), формируемый на основе значений старших битов адреса (табл. 5.4).

---

[1] *Hinden R., Deering S.* Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513. URL: `http://www.faqs.org/rfcs/rfc3513.html`.

[2] *Blanchet M.* A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block, RFC 3531. URL: `http://www.faqs.org/rfcs/rfc3531.html`.

[3] *Hinden R., Deering S., Nordmark E.* IPv6 Global Unicast Address Format, RFC 3587. URL: `http://www.faqs.org/rfcs/rfc3587.html`.

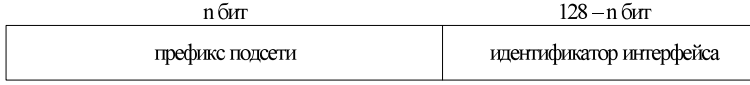


Рис. 5.12. Префикс в структуре адреса IPv6

Таблица 5.4

Префиксы адресов IPv6

Назначение	Префикс (двоичный)
Зарезервировано	0000 0000
Не определено	0000 0001
Зарезервировано для NSAP	0000 001
Зарезервировано для IPX	0000 010
Не определено	0000 011
Не определено	0000 1
Не определено	0001
Не определено	001
Провайдерские Unicast-адреса	010
Не определено	011
Зарезервировано для географических уникаст-адресов	100
Не определено	101
Не определено	110
Не определено	1110
Не определено	1111 0
Не определено	1111 10
Не определено	1111 110
Не определено	1111 1110 0
Адреса локальной связи	1111 1110 10
Адрес локальной подсети	1111 1110 11
Адреса группы интерфейсов (Multicast-адреса)	1111 1111

Префикс представляет собой часть адреса с битами фиксированного значения или битами, служащими идентификатором сети. Запись длины префикса адреса IPv6 аналогична нотации CIDR для IPv4:

`|адрес/длина_префикса|`.

Примеры указания префикса: `21DA:D3:0:2F3B::/64`, `21DA:D3::/48`.

Типы адресов

Адрес IPv6 идентифицирует интерфейс, а не конкретный узел. Интерфейс может ассоциироваться только с одним узлом, но на узле может быть несколько интерфейсов, каждому из которых можно поставить в соответствие несколько IPv6-адресов.

Выделяют следующие типы адресов IPv6:

- *адрес одиночного интерфейса (Unicast)*;
- *адрес любого интерфейса группы интерфейсов (Anycast)*;
- *адрес группы интерфейсов (Multicast)*.

Адрес одиночного интерфейса. Этот тип адресов применяется для идентификации только одного интерфейса.

Выделяют несколько форм этого типа адреса:

- глобальный адрес одиночного интерфейса провайдера (Global Provider Based Unicast Address);
- адрес локальной связи (Link-Local-Use Address);
- адрес локальной подсети (Site-Local-Use Address);
- адреса совместимости с адресами IPv4;
- адрес NSAP (Network Service Access Point).

В *глобальном адресе одиночного интерфейса провайдера* первые три бита являются префиксом и установлены в `001`. Такой тип адреса эквивалентен общедоступному адресу IPv4 и используется для глобальной маршрутизации в части Интернет.

На рис. 5.13 и 5.14 представлены форматы глобального Unicast-адреса и глобального адреса одиночного интерфейса провайдера соответственно.

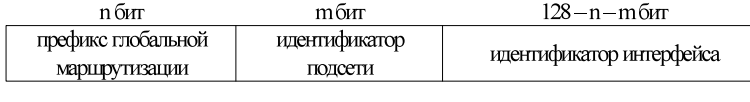


Рис. 5.13. Общая структура глобального Unicast-адреса IPv6

Поле *Агрегированный идентификатор верхнего уровня (TLA ID, Top Level Aggregation Identifier* имеет длину 13 бит, что позволяет создать 8 192 различных идентификаторов TLA. Здесь указывается

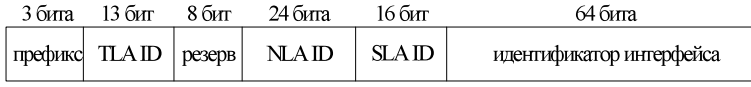


Рис. 5.14. Структура глобального Unicast-адреса провайдера

идентификатор самого верхнего уровня маршрутизации. За распределение адресного пространства этого уровня отвечает *IANA (Internet Assigned Numbers Authority)*. Адреса маршрутизаторов верхнего уровня маршрутизации имеют 16-битные префиксы, соответствующие выделенным крупным поставщикам услуг Интернета агрегированным идентификаторам верхнего уровня.

Далее идёт зарезервированное поле *Резерв (Res)* длиной 8 бит.

Поле *Агрегированный идентификатор следующего уровня (NLA ID, Next Level Aggregation Identifier)* длиной 24 бита содержит идентификатор элемента следующего уровня маршрутизации, т.е. указывает подсеть поставщика услуг Интернета, которую он может выделить для какой-то крупной организации.

Поле *Агрегированный идентификатор уровня подсети (SLA ID, Site Level Aggregation Identifier)* длиной 16 бит содержит идентификатор элемента уровня подсети, т.е. указывает подсеть в пределах адресного пространства организации.

Поле *Идентификатор интерфейса (Interface ID)* длиной 64 бита содержит идентификатор конкретного интерфейса узла сети.

Адрес локальной связи (Link-Local-Use Address) имеет префикс `11111110 10` и по смыслу эквивалентен адресам APIPA (Automatic Private IP Addressing) `169.254.0.0/16`, используемым в IPv4. На рис. 5.15 приведён формат адреса.

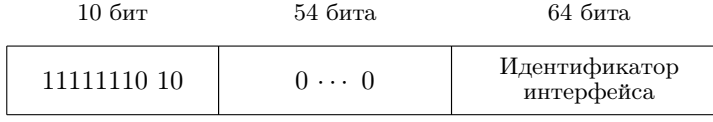


Рис. 5.15. Структура адреса локальной связи IPv6

В адресах локальной связи первые 64 бита фиксированы, т.е. префикс имеет длину 64: `FE80::/64`, остальные 64 бита идентифицируют конкретный интерфейс узла (поле *Идентификатор интерфейса (Interface ID)*).

Адрес локальной подсети (Site-Local-Use Address) имеет префикс `11111110 11`. Этот тип адресов эквивалентен частному адресному

пространству 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16 в IPv4. На рис. 5.16 приведён формат адреса локальной подсети.

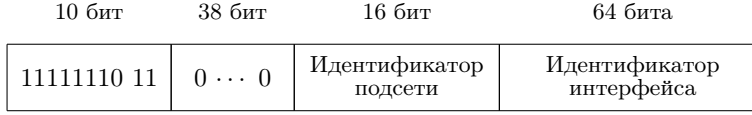


Рис. 5.16. Структура адреса локальной подсети IPv6

В адресах локальной подсети первые 48 бит фиксированы, т.е. префикс имеет длину 48: FEC0::/48.

Следующие 16 бит идентифицируют подсеть (*Subnet ID*) организации.

Поле *Идентификатор интерфейса (Interface ID)* длиной 64 бита содержит идентификатор конкретного интерфейса узла подсети.

Для перехода от IPv4-адресов к адресам IPv6 используются *адреса совместимости*:

– *IPv4-совместимый адрес* — IPv6-адрес, имеющий следующую структуру:

  0:0:0:0:0:0:w.x.y.z или ::w.x.y.z,

  где w.x.y.z — IPv4-адрес в десятично-точечном представлении. Пример IPv4-совместимого адреса — ::80.162.174.25.
– *IPv4-сопоставленный адрес* — адрес IPv4, преобразованный к виду IPv6 и имеющий следующий формат

  0:0:0:0:0:FFFF:w.x.y.z или ::FFFF:w.x.y.z,

  где w.x.y.z — IPv4-адрес в десятично-точечном представлении. Пример IPv4-сопоставленного адреса: ::FFFF:206.62.226.33.
– *Адрес 6to4* используется узлами, работающими и с IPv4, и с IPv6. Чтобы его получить, 32-битный адрес IPv4 переводится в двоичную форму и присоединяется к префиксу 2002::/16. Например, IPv4-адресу 131.107.0.1 будет соответствовать IPv6-адрес 2002:836B:1::/48.

Адреса NSAP (Network Service Access Point) имеют префикс формата 0000001, а последние 121 бит адреса IPv6 сопоставляются адресу NSAP (см. RFC-1888[1]).

Адрес любого интерфейса группы интерфейсов. Этот тип адреса используется для идентификации некоторой группы интерфейсов. Доставка пакетов производится на один из адресов, принадлежащих группе, соответствующей идентификатору.

[1]OSI NSAPs and IPv6, RFC 1888 / J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd. URL: http://www.faqs.org/rfcs/rfc1888.html.

На данный момент такие адреса назначаются только маршрутизаторам из пространства адресов одиночных интерфейсов и используются для взаимодействия с одним из нескольких маршрутизаторов некоторой подсети. Адрес формируется на базе префикса подсети с занулением остальных бит, идентифицирующих конкретный интерфейс (рис. 5.17).

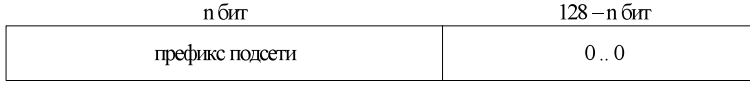


Рис. 5.17. Структура Anycast-адреса IPv6

Всем интерфейсам маршрутизатора, подключённым к подсети, назначается этот же адрес для соответствующей подсети.

Адрес группы интерфейсов. Адрес группы интерфейсов () используется для многоадресной рассылки (маршрутизации).

Его формат приведён на рис. 5.18.

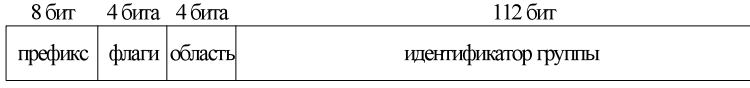


Рис. 5.18. Структура глобального Multicast-адреса провайдера

Адресу соответствует *префикс* `11111111`.

После префикса идут *Флаги (Flags)*, занимающие 4 бита. В соответствии с RFC-3513[1] и RFC-2373 первые три бита выставлены в ноль. Последний бит указывает, является адрес временным (значение 1) или постоянно назначенным IANA (значение 0).

После флагов указывается *Область (Scope)* объединённой сети IPv6, в которую направляется трафик. Значение этого поля определяет допустимый предел многоадресной рассылки. В документах RFC-3513[2] и RFC-2373 определены следующие значения данного поля:

– значения 0, 3, F зарезервированы;
– значение 1 — область локального интерфейса (interface-local scope);
– значение 2 — область локальной связи (link-local scope);

[1] *Hinden R., Deering S.* Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513. URL: `http://www.faqs.org/rfcs/rfc3513.html`.

[2] *Hinden R., Deering S.* Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513. URL: `http://www.faqs.org/rfcs/rfc3513.html`.

– значение 4 — область локального администрирования (admin-local scope);
– значение 5 — область локальной подсети (site-local scope);
– значение 8 — область локальной подсети организации (organization-local scope);
– значение E — неограниченная область (global scope);
– значения 6, 7, 9, A, B, C, D не определены.

Поле *Идентификатор группы (Group ID)* длиной 112 бит собственно определяет группу многоадресной рассылки. Постоянно назначенные идентификаторы групп не зависят от значения поля Scope (Область). Адреса с префиксами от `FF01::` до `FF0F::` зарезервированы.

Адрес `FF01::1` является многоадресным адресом внутри локального узла, `FF02::1` — многоадресным адресом всех узлов локальной связи, `FF01::2` — многоадресным адресом внутри маршрутизатора, `FF02::2` — многоадресным адресом всех маршрутизаторов локальной связи и, наконец, `FF05::2` — многоадресным адресом всех маршрутизаторов локальной подсети.

Если в идентификаторе группы последние 32 бита имеют значение, а предыдущие 80 бит выставлены в ноль, то каждый идентификатор группы сопоставляется с уникальным MAC-адресом многоадресной рассылки Ethernet.

Адрес запроса узлов (Solicited-node Multicast Address) функционально является одновременно адресом одиночного интерфейса и Anycast-адресом. Адрес имеет префикс `FF02::1:FF00:0/104`, а последние биты выступают в качестве Unicast или Anycast-адреса IPv6.

Например, узлу с адресом IPv6 локальной связи

`FE80::2AA:FF:FE28:9C5A`

соответствует адрес запроса узлов

`FF02::1:FF28:9C5A.`

Адреса серверов и маршрутизаторов

Узел IPv4 с одним сетевым адаптером обычно имеет один адрес IPv4. Узел же IPv6 обычно имеет несколько адресов IPv6, даже при наличии только одного интерфейса.

Узлу IPv6 назначаются следующие Unicast-адреса:
– адрес локальной связи для каждого интерфейса;
– адреса для каждого интерфейса (это может быть адрес локальной подсети и один или несколько глобальных адресов);
– адрес замыкания на себя (Loopback Address).

Кроме того, каждый узел прослушивает трафик на следующих адресах многоадресной рассылки:
– адрес всех узлов в области локального узла (`FF01::1`);
– адрес всех узлов в области локальной связи (`FF02::1`);

– адрес запроса узла для каждого Unicast-адреса на каждом интерфейсе;
– адреса многоадресной рассылки для групп, присоединённых к каждому интерфейсу.

Маршрутизатору IPv6 назначаются следующие Unicast-адреса:
– адрес локальной связи для каждого интерфейса;
– Unicast-адреса для каждого интерфейса (это может быть адрес локальной подсети и один или несколько глобальных адресов);
– адрес замыкания на себя (Loopback Address).

Маршрутизатору IPv6 назначаются следующие Anycast-адреса:
– Anycast-адрес для каждой подсети;
– дополнительные Anycast-адреса (необязательно).

Кроме того, каждый маршрутизатор прослушивает трафик на следующих адресах многоадресной рассылки:
– адрес всех узлов в области локального узла (`FF01::1`);
– адрес всех маршрутизаторов в области локального узла (`FF01::2`);
– адрес всех узлов в области локальной связи (`FF02::`);
– адрес всех маршрутизаторов в области локальной связи (`FF02::2`);
– адрес всех маршрутизаторов в области локальной подсети (`FF05::2`);
– адрес запроса узла для каждого Unicast-адреса на каждом интерфейсе;
– адреса групп, присоединённых к каждому интерфейсу.

Идентификаторы интерфейса IPv6

Последние 64 бита адреса IPv6 представляют собой идентификатор интерфейса, уникальный для 64-битного префикса адреса IPv6. Идентификатор интерфейса может определяться одним из следующих способов:
– согласно RFC-3513[1] (ранее RFC-2373), все Unicast-адреса, имеющие префиксы с 001 по 111, должны также использовать 64-битный идентификатор интерфейса, образованный из адреса *EUI-64 (Extended Unique Identifier)*;
– документ RFC-3041[2] описывает генерируемый случайным образом идентификатор интерфейса, изменяющийся со временем для обеспечения определённого уровня анонимности;
– идентификатор интерфейса, назначаемый при автоматической настройке адреса с ведением базы данных (например, по протоколу DHCPv6);

[1] *Hinden R., Deering S.* Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513. URL: `http://www.faqs.org/rfcs/rfc3513.html`.

[2] *Narten T., Draves R.* Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC 3041. URL: `http://www.faqs.org/rfcs/rfc3041.html`.

– идентификатор интерфейса, настраиваемый вручную.

Сопоставление адресов EUI-64 с идентификаторами интерфейсов IPv6. Адреса IEEE EUI-64 представляют новый стандарт адресации сетевых интерфейсов. Длина идентификатора компании по-прежнему составляет 24 бита, но идентификатор расширения имеет длину 40 бит, что создаёт гораздо большее пространство адресов для изготовителей сетевых адаптеров. Адрес EUI-64 использует биты U/L и I/G точно так же, как адрес IEEE 802.

Адрес EUI-64 из адреса IEEE 802 образуется путём вставки между идентификатором компании и идентификатором расширения в адресе IEEE 802 следующих 16 бит: `11111111 11111110` (`0xFF 0xFE`).

64-битный идентификатор интерфейса для Unicast-адресов IPv6 создаётся путём инвертирования бита U/L в адресе EUI-64 (значение 1 обращается в 0; значение 0 обращается в 1). Для получения идентификатора интерфейса IPv6 из адреса IEEE 802 необходимо сначала сопоставить этот адрес IEEE 802 с адресом EUI-64, а затем инвертировать бит U/L.

Пример 1. Преобразование адреса IEEE 802.

Узел имеет MAC-адрес Ethernet `00:AA:00:3F:2A:1C`. Сначала этот адрес преобразуется в формат EUI-64 путём вставки разрядов `FF:FE` между третьим и четвёртым байтами:

`00:AA:00:FF:FE:3F:2A:1C`.

Затем инвертируется бит U/L (седьмой бит в первом байте). Первый байт в двоичной форме имеет вид `00000000`. При инвертировании седьмого бита он принимает вид `00000010` (`0x02`). Конечный результат

`02:AA:00:FF:FE:3F:2A:1C`

после преобразования в двухточечно-шестнадцатеричную нотацию становится идентификатором интерфейса: `2AA:FF:FE3F:2A1C`. Таким образом, сетевому адаптеру с MAC-адресом `00:AA:00:3F:2A:1C` соответствует адрес локальной связи `FE80::2AA:FF:FE3F:2A1C`.

Идентификаторы интерфейса временного адреса. Пользователь Интернета при подключении к поставщику услуг Интернета получает адрес IPv4 с использованием протоколов *PPP (Point-to-Point Protocol)* и *IPCP (Internet Protocol Control Protocol)*. Адрес IPv4 может изменяться при каждом подключении данного пользователя. В связи с этим сложно отследить трафик пользователя в Интернете по IP-адресу.

При подключении удалённого доступа на базе протокола IPv6 после установки соединения пользователю назначается 64-битный префикс (путём обнаружения маршрутизатора и автонастройки адреса). Если идентификатор интерфейса всегда основан на адресе EUI-64 (полученном из статического адреса IEEE 802), то существует возможность

определения трафика конкретного узла независимо от его префикса, что облегчает отслеживание конкретных пользователей. Для обеспечения определённого уровня анонимности в документе RFC-3041[1] описан альтернативный идентификатор интерфейса IPv6, генерируемый случайным образом и изменяемый с течением времени.

Исходный идентификатор интерфейса генерируется с использованием случайных чисел. Для систем IPv6, не способных хранить протокольные сведения для создания будущих значений идентификатора интерфейса, при каждой инициализации протокола IPv6 генерируется новый случайный идентификатор интерфейса. Для систем IPv6, имеющих возможность хранения протокольных сведений, при инициализации протокола IPv6 новый идентификатор интерфейса создаётся следующим образом:

1. загружается значение из хранилища протокольных сведений и добавляется к идентификатору интерфейса, основанному на адресе EUI-64 адаптера;
2. для величины, полученной на шаге 1, вычисляется хеш-функция MD5;
3. последние 64 бита хеша MD5, вычисленного на шаге 2, сохраняются для вычисления следующего идентификатора интерфейса;
4. для седьмого из первых 64 бит хеша MD5, вычисленного на шаге 2, задаётся значение 0 (седьмым является бит U/L, нулевое значение которого задаёт локальное администрирование идентификатора интерфейса), результат которого представляет собой идентификатор интерфейса.

Итоговый адрес IPv6, основанный на таком случайном идентификаторе интерфейса, называют *временным адресом*. Временные адреса создаются для префиксов общедоступных адресов, использующих автонастройку адресов без ведения базы данных.

5.3. Другие протоколы межсетевого уровня стека TCP/IP

5.3.1. Протокол ICMP

Протокол передачи команд и сообщений об ошибках (Internet Control Message Protocol, ICMP) используется программным обеспечением ЭВМ при взаимодействии друг с другом в рамках идеологии TCP/IP (см. RFC 792[2]).

[1] *Narten T., Draves R.* Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC 3041. URL: `http://www.faqs.org/rfcs/rfc3041.html`.

[2] *Postel J.* Internet Control Message Protocol, RFC 792. URL: `http://www.faqs.org/rfcs/rfc792.html`.

Функциональность протокола ICMP

ICMP-протокол осуществляет:
- передачу отклика на пакет или эхо на отклик;
- контроль времени жизни дейтаграмм в системе;
- реализует переадресацию пакета;
- выдаёт сообщения о недостижимости адресата или о некорректности параметров;
- формирует и пересылает временные метки;
- выдаёт запросы и отклики для адресных масок и другой информации.

ICMP-сообщения об ошибках никогда не выдаются в ответ на:
- ICMP-сообщение об ошибке;
- при мультикастинге или широковещательной адресации;
- для фрагмента дейтаграммы (кроме первого);
- для дейтаграмм, чей адрес отправителя является нулевым, широковещательным или мультикастинговым.

Эти правила призваны блокировать потоки дейтаграмм, посылаемых в ответ на мультикастинг или широковещательные ICMP-сообщения.

Форматы пакетов ICMP

ICMP-сообщения могут быть нескольких типов. Поэтому все ICMP-пакеты начинаются с 8-битного поля *Тип ICMP* и поля *Код* (15 значений).

Типы ICMP:

0 — эхо-ответ (ping-отклик);
3 — адресат не достижим;
4 — отключение источника при переполнении очереди;
5 — изменить маршрут;
8 — эхо-запрос (ping-запрос);
9 — объявление маршрутизатора;
10 — запрос маршрутизатора;
11 — для дейтаграмм время жизни истекло (TTL=0);
12 — проблема с параметрами дейтаграммы;
13 — запрос временной метки;
14 — временная метка-отклик;
15 — запрос информации;
16 — информационный отклик;
17 — запрос адресной маски;
18 — отклик на запрос адресной маски.

Код уточняет функцию ICMP-сообщения (например, код 1 в типе ICMP 3 указывает на недостижимость ЭВМ, а код 12 для того же типа — на недоступность ЭВМ для данного вида сервиса).

На рис. 5.19 приведён формат эхо-запроса и отклика ICMP.

Рис. 5.19. Формат эхо-запроса и отклика ICMP

Поля *Идентификатор* (16 бит) и *Номер по порядку* (16 бит) служат для того, чтобы отправитель мог связать в пары запросы и отклики.

Поле *Тип* определяет, является ли пакет запросом (Тип=8) или откликом (Тип=0).

Поле *Контрольная сумма* представляет собой 16-разрядное дополнение по модулю 1 контрольной суммы всего ICMP-сообщения, начиная с поля *Тип*.

Поле *Данные* служит для записи информации, возвращаемой отправителю. Размер данного поля не регламентирован и определяется предельным размером IP-пакета.

Сообщение «адресат не достижим» посылается в случае, если маршрутизатор не может доставить дейтаграмму по назначению. На рис. 5.20 приведён формат ICMP-сообщения «адресат не достижим».

Рис. 5.20. Формат ICMP-сообщения «адресат не достижим»

Поле *MTU на следующем этапе* характеризует максимальную длину пакетов на очередном шаге пересылки.

По полю *Internet-заголовок (включая опции) + первые 64 бита дейтаграммы* можно определить, какой адрес оказался недостижимым.

В ситуации, когда принимающая сторона не справляется с приёмом потока, отправителю может быть послано сообщение с требованием

снижения нагрузки. На рис. 5.21 приведён формат ICMP-запроса снижения загрузки.

Рис. 5.21. Формат ICMP-запроса снижения загрузки

Если маршрутизатор обнаружит, что станция использует неоптимальный маршрут, он может послать ей ICMP-запрос о переадресации. Команду переадресации маршрутизатор посылает только станциям, но не маршрутизаторам. На рис. 5.22 приведён формат ICMP-запроса переадресации.

Рис. 5.22. Формат ICMP-запроса переадресации

Поле *Internet-адрес маршрутизатора* содержит адрес маршрутизатора, который станция должна использовать для отправки дейтаграммы по указанному в заголовке месту назначения.

В поле *Internet-заголовок (включая опции) + первые 64 бита дейтаграммы* кроме самого заголовка расположены первые 64 бита дейтаграммы, вызвавшей это сообщение.

Маршрутные таблицы формируются в результате запросов и объявлений, посылаемых маршрутизаторами. Когда в сети появляется новый маршрутизатор, он посылает широковещательный запрос. В ответ другие маршрутизаторы сети посылают сообщения об имеющихся маршрутах. На рис. 5.23 приведён формат ICMP-запроса об имеющихся маршрутах.

Рис. 5.23. Формат ICMP-запроса об имеющихся маршрутах

Поле *Число адресов* указывает число адресных записей в сообщении.

Поле *Длина адреса* содержит число 32-битных слов, необходимых для описания адреса маршрутизатора.

Поле *Время жизни* указывает продолжительность жизни объявленных маршрутов в секундах.

Поле *Уровень приоритета* указывает приоритет маршрута по отношению к другим маршрутам данной подсети.

На рис. 5.24 приведён формат ICMP-запроса маршрутной информации. На рис. 5.25 приведён формат ICMP-запроса (отклика) маски подсети, в котором поле *Адресная маска* содержит 32-разрядную маску подсети.

Рис. 5.24. Формат ICMP-запроса маршрутной информации

При ликвидации пакета по истечении TTL маршрутизатор посылает отправителю сообщение «время истекло». На рис. 5.26 приведён формат ICMP-сообщения «время (TTL) истекло».

На рис. 5.27 приведён формат ICMP-сообщения «конфликт параметров», посылаемого маршрутизатором при выявление какой-либо ошибки (не из числа описанных выше).

Рис. 5.25. Формат ICMP-запроса (отклика) маски подсети

Рис. 5.26. Формат ICMP-сообщения «время (TTL) истекло»

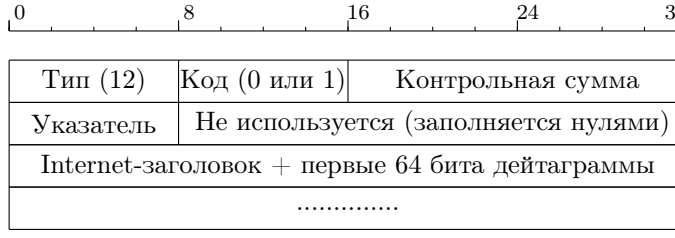


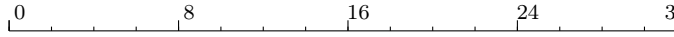
Рис. 5.27. Формат ICMP-сообщения типа «конфликт параметров»

Поле *Указатель* отмечает октет дейтаграммы, из-за которого возникла ошибка.

В процессе трассировки маршрутов может возникнуть проблема синхронизации времени на различных станциях. В этом случае делается запрос временно́й метки. На рис. 5.28 приведён формат ICMP-запроса временно́й метки.

Поле *Тип* со значением 13 указывает, что это запрос, а тип 14 — на то, что это отклик.

Поля *Идентификатор* (16 бит) и *Номер по порядку* (16 бит) служат



Тип (13 или 14)	Код (0)	Контрольная сумма
Идентификатор		Номер по порядку
Исходная временная метка		
Временная метка на входе		
Временная метка на выходе		

Рис. 5.28. Формат ICMP-запроса временной метки

для того, чтобы отправитель мог связать в пары запросы и отклики.

Поле *Исходная временна́я метка* заполняется отправителем непосредственно перед отправкой пакета.

Поле *Временна́я метка на входе* заполняется маршрутизатором при получении данного пакета.

Поле *Временна́я метка на выходе* заполняется маршрутизатором непосредственно перед отправкой данного пакета.

5.3.2. Протокол ARP

Преобразование IPv4-адресов (4 байта), задаваемых с учётом положения узла в сети, в MAC-адреса (6 байт для Ethernet), заданные аппаратным образом, выполняется с помощью так называемой *ARP-таблицы* (см. RFC 826[1]). Каждый узел сети имеет отдельную ARP-таблицу для каждого своего сетевого адаптера. Протокол *ARP (Address Resolution Protocol)* преобразует ARP-адреса в Ethernet-адреса.

Процедура преобразования адресов

При обмене сообщениями между двумя прикладными программами для определения Ethernet-адреса просматривается ARP-таблица. Если для требуемого IP-адреса в ARP-таблице присутствует Ethernet-адрес, то формируется и посылается соответствующий пакет. В противном случае выполняются следующие действия:

1. всем узлам в сети посылается пакет с ARP-запросом (с широковещательным Ethernet-адресом места назначения), а исходящий IP-пакет ставится в очередь;

[1] *Plummer D. C.* Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, RFC 826. URL: `http://www.faqs.org/rfcs/rfc826.html`.

2. каждый узел, принявший ARP-запрос, в своём ARP-модуле сравнивает собственный IP-адрес с IP-адресом в запросе:
   – если IP-адрес совпал, то по Ethernet-адресу отправителя запроса посылается ответ, содержащий как IP-адрес ответившего узла, так и его Ethernet-адрес, а в ARP-таблице узла-отправителя формируется соответствующий элемент и отправляется IP-пакет, ранее поставленный в очередь;
   – если же в сети нет узла с искомым IP-адресом, то ARP-ответа не будет и не будет записи в ARP-таблице, а протокол IP уничтожит IP-пакеты, предназначенные этому адресу.

Формат пакета ARP

Формат пакета ARP представлен на рис. 5.29.

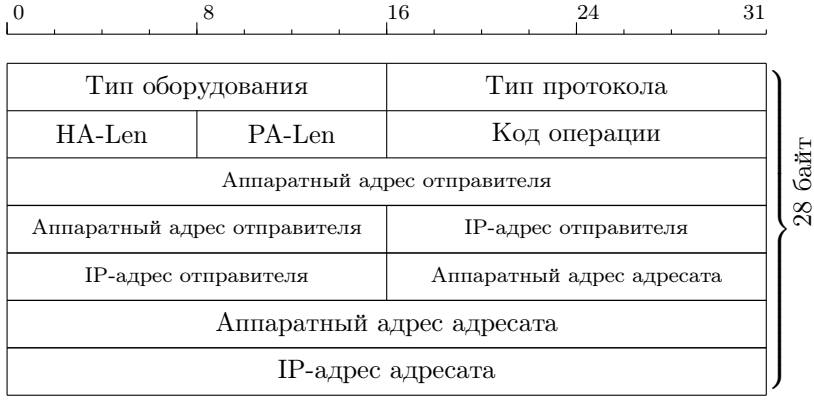


Рис. 5.29. Формат заголовка пакета ARP

Поле *Тип оборудования* (16 бит) указывает код типа интерфейса, для которого отправитель ищет адрес.

Поле *Тип протокола* (16 бит) содержит код типа протокола (например, код IP-протокола имеет значение $0800H$, код ARP-протокола — $0806H$, код RARP-протокола — $8035H$, код SNMP-протокола — $814CH$).

Поле *HA-Len* (8 бит) указывает длину аппаратного адреса.

Поле *PA-Len* (8 бит) указывает длину протокольного адреса в байтах (например, для IP-адреса *PA-Len=4*).

Поле *Код операции* (16 бит) определяет, является ли данный пакет ARP-запросом (код = 1), ARP-откликом (код = 2), RARP-запросом (код = 3) или RARP-откликом (код = 4).

Остальные поля определяют соответственно аппаратный адрес отправителя, IP-адрес отправителя, аппаратный адрес адресата, IP-адрес адресата.

5.3.3. Протокол RARP

Протокол *RARP (Reverse Address Resolution Protocol)* предназначен для обратной трансляции адресов, т.е. для преобразования MAC-адресов в IP-адреса (см. RFC 903[1]).

Протокол RARP предполагает наличие специального сервера, обслуживающего RARP-запросы и хранящего базу данных о соответствии аппаратных адресов протокольным адресам.

Формат пакета RARP

Протокол RARP имеет сходный с ARP формат сообщения (рис. 5.30).

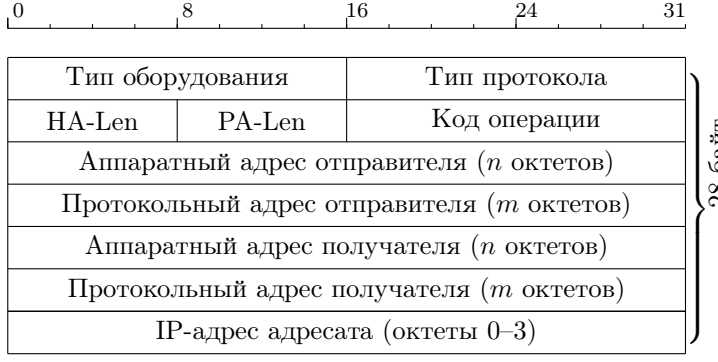


Рис. 5.30. Формат RARP-сообщения

Поле *Тип оборудования* (16 бит) указывает тип интерфейса, для которого отправитель ищет адрес (например, для Ethernet код имеет значение 1).

Поле *Тип протокола* (16 бит) содержит код типа протокола (например, код IP-протокола имеет значение $0800H$, код ARP-протокола — $0806H$, код RARP-протокола — $8035H$, код SNMP-протокола — $814CH$).

[1] A Reverse Address Resolution Protocol, RFC 903 / R. Finlayson, T. Mann, J. Mogul, M. Theimer. URL: `http://www.faqs.org/rfcs/rfc903.html`.

Поле *HA-Len* (8 бит) указывает длину аппаратного адреса (задаёт значение n).

Поле *PA-Len* (8 бит) указывает длину протокольного адреса в байтах (задаёт значение m; например, для IP-адреса *PA-Len=4*).

Поле *Код операции* (16 бит) определяет, является ли данный пакет ARP-запросом (код = 1), ARP-откликом (код = 2), RARP-запросом (код = 3) или RARP-откликом (код = 4).

Применение протокола RARP

Протокол RARP применяется, например, когда необходимо инициализировать бездисковую рабочую станцию (так как нет возможности сохранять IP-адрес на жёстком диске): для переноса из сервера в память образа операционной системы может использоваться протокол TFTP, при этом IP-адреса сервера и станции-клиента должны быть известны.

5.4. Маршрутизация

Процесс маршрутизации можно разделить на два иерархически связанных уровня:
- уровень маршрутизации,
- уровень передачи пакетов.

На уровне маршрутизации происходит работа с таблицей маршрутизации. Таблица маршрутизации служит для определения адреса (сетевого уровня) следующего маршрутизатора или непосредственно получателя по имеющемуся адресу (сетевого уровня). После определения адреса передачи выбирается определённый выходной физический порт маршрутизатора. Этот процесс называется *определением маршрута перемещения пакета*. Настройка таблицы маршрутизации осуществляется *протоколами маршрутизации (Routing Protocols)*. Примерами протоколов маршрутизации являются протоколы RIP, OSPF, BGP и др.

Уровень передачи пакетов обрабатывает команды, поступающие с уровня маршрутизации. Перед передачей пакета на этом уровне проверяется контрольная сумма заголовка пакета, определяется адрес (канального уровня) получателя пакета и производится отправка пакета с учётом очерёдности, фрагментации, фильтрации и т.д. На данном уровне используются протоколы называемые *сетевыми протоколами (Routed Protocols)*, к которым можно отнести, например, протоколы IP, IPX, AppleTalk.

Таким образом, служебная информация протоколов маршрутизации вкладывается в пакет сетевого уровня, формированием которого занимается сетевой протокол.

5.4.1. Ядерная маршрутизация

Маршрутизатор может быть реализован либо полностью *программным способом* (в этом случае он представляет собой модуль операционной системы, установленной на компьютере общего назначения, выполняющем функции сервера), либо *аппаратно-программным способом* (является специализированным вычислительным устройством, в котором часть функций выполняется нестандартной аппаратурой, а часть — программными модулями, работающими под управлением специализированной операционной системы, называемой *монитором*).

Основное преимущество программных маршрутизаторов перед аппаратными — гибкость, интеллектуальность и простота модификации алгоритмов. Возможны реализации самых нестандартных сетевых решений на базе программного маршрутизатора.

Большая часть современных программных маршрутизаторов функционирует под управлением ОС Linux, что позволяет обеспечить высокую производительность и гибкость конфигураций при осуществлении маршрутизации, а также предоставляет широкие возможности по обработке сетевого трафика, поступающего на физический интерфейс маршрутизатора.

Основная функциональность Linux-маршрутизатора обеспечивается ядром операционной системы. Любая ОС Linux, начиная с версии ядра 2.2, содержит обновлённую сетевую подсистему, архитектура которой была значительно пересмотрена и перестроена. Как результат — функциональность, превосходящая возможности аппаратно-программных маршрутизаторов, что позволяет реализовывать разнообразное управление сетевым трафиком, в частности, накладывать ограничения на транзитный трафик и осуществлять маршрутизацию на основе как идентификатора пользователя, адреса назначения, номера порта соединения, типа сервиса и других полей сетевых заголовков, так и непосредственного содержания передаваемых в пакетах данных.

В большинстве случаев настройка системы маршрутизации не представляет особых сложностей, поскольку большинство операций выполняется ядром и специализированными программными пакетами автоматически. ОС Linux сильна как раз своим сетевым инструментарием, поэтому данную систему можно использовать в качестве связующего звена даже в тех локальных и глобальных сетях, основную часть которых составляют компьютеры, работающие под управлением Windows, MacOS и др.

Iproute2

Ранее подсистема маршрутизации в Linux, как и большинство операционных систем UNIX, использовала утилиты arp, ifconfig и route. Но, начиная с ядра 2.2, сетевая подсистема была полностью

переписана. Новый сетевой код дал увеличение производительности и более высокие эксплуатационные характеристики.

Современная реализация маршрутизации в ядре Linux основана на подсистеме iproute2. Управление на прикладном уровне представлено пакетом IProute2, входящим в большинство дистрибутивов Linux[1].

Фактически iproute2 состоит из нескольких утилит управления трафиком:

- `ip` — управление маршрутизацией;
- `tc` — управление очередями маршрутизации;
- `ss` — просмотр текущих соединений и открытых портов.

Утилита «ip» заменяет собой команды route, arp, ifconfig и предназначена для управления таблицами маршрутизации, в частности, правилами, определёнными в них, и помогает реализовывать возможности многотабличной маршрутизации, туннелирования, а также многоадресную маршрутизацию.

Кроме того, Linux имеет гибкую систему управления трафиком, называемую *Traffic Control*. Эта система поддерживает множество методов для классификации, приоритезации, разделения и ограничения как входящего трафика, так и исходящего. Управление этими функциями осуществляет вторая утилита пакета — «tc». Эта утилита также позволяет реализовать QoS в нужном для системы объёме:

- разделение разных типов трафика по классам (не только по битам ToS в IP-пакете, но и по другим данным из заголовка IP-пакета);
- назначение различных дисциплин обработки очередей трафика с разным приоритетом, механизмами прохождения очереди, ограничениями по скорости и т.п.

Обе утилиты интерпретируют выполнение команд, а все функции выполняет ядро.

Принципиальной особенностью iproute2 является использование нескольких таблиц маршрутизации. Когда ядру необходимо выбрать маршрут, оно определяет, в соответствии с какой таблицей это нужно делать. Простейшим примером использования нескольких таблиц маршрутизации является подключение сети через двух (или более) провайдеров (рис. 5.31). Такое подключение может требоваться для обеспечения отказоустойчивости или балансировки нагрузки.

Другой важной особенностью iproute2 является возможность организации туннелей. В Linux поддерживаются 3 типа туннелирования — IP-в-IP, GRE-туннелирование и туннелирование не-ядерного уровня (например, PPTP).

Туннели IP-в-IP являются самыми простыми, однако имеют ряд ограничений. Например, их организация возможна только в сетях на базе протокола IPv4. Кроме того, отсутствует интероперабельность с

[1]При этом старые команды (ifconfig, route) используют новую подсистему с некоторыми параметрами по-умолчанию.

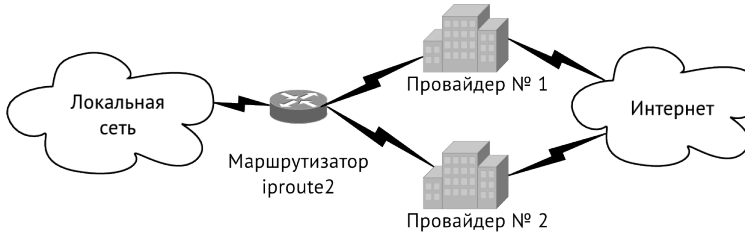


Рис. 5.31. Подключение через двух провайдеров

другими операционными системами. При этом туннели IP-в-IP могут широко применяться, например, при виртуализации.

GRE — стандартный протокол туннелирования, разработанный фирмой Cisco. В отличие от туннелей IP-в-IP, он поддерживает широковещательные сообщения и может работать в сетях на базе протокола IPv6.

Кроме маршрутизации iproute2 может осуществлять управление обработкой пакетов и, в частности, дисциплинами обработки очередей (как бесклассовыми, так и на основе классов). Также iproute2 может проводить маркировку пакетов (т.е. выполнять роль классификатора).

Несмотря на все преимущества, iproute2 имеет несколько недостатков:

1) неполная документированность;
2) маршрутизатор конфигурируется интерактивно, т.е. путём ввода команд с клавиатуры;
3) из-за своей монолитной архитектуры iproute2 имеет более высокую сложность, чем, например, Click.

Click

Click Modular Router [56] представляет собой специализированное программное обеспечение для создания высокопроизводительных программных маршрутизаторов. Click был разработан в Массачусетском технологическом университете США при поддержке национального агентства DARPA.

Маршрутизатор Click имеет модульную структуру. Отдельные элементы осуществляют простые функции маршрутизатора, такие, как классификация пакетов, организация очередей, планирование, установление связи с сетевыми устройствами. Каждый элемент представлен C++ объектом.

Конфигурация программного маршрутизатора может быть представлена в виде направленного графа с различными элементами в

качестве вершин (рис. 5.32), в котором пакеты с данными перемещаются вдоль рёбер.

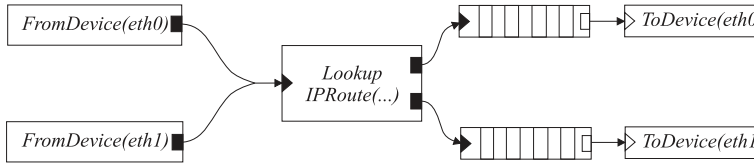


Рис. 5.32. Конфигурационный граф стандартного маршрутизатора

Благодаря архитектуре программного средства и декларативному языку описания конфигурации маршрутизатор является модульным и легко расширяемым. Высокая скорость обработки трафика достигается посредством механизма *Device Polling* (технологии работы ядра Linux с устройствами), а также за счёт внутреннего механизма аннотаций.

Как было сказано, Click состоит из набора модулей с единой системой конфигурации. Каждый модуль реализует простые функции (классификацию пакетов, управление очередями, функции планировщика и интерфейса с сетевыми устройствами) и представляет собой отдельную часть процесса маршрутизации. Модуль может выполнять как простые вычисления (например, уменьшение счётчика жизни IP-пакета), так и более сложные (построение маршрута следования пакета).

Для построения маршрутизатора выбирается набор обработчиков, которые соединяются в ориентированный граф. Обработчик Click представляет собой некий элемент, в котором происходит обработка пакета. Click позволяет создавать новые обработчики, а также модифицировать и комбинировать между собой имеющиеся. Действия маршрутизатора задаются при помощи набора определённых модулей, а также путём определения связей между ними.

Для каждого элемента маршрутизатора должны быть определены:

- *класс (Element Class)*, задающий действия элемента маршрутизатора при приёме пакета;
- *порты (Ports)* для создания соединения между элементами;
- *конфигурационная строка (Configuration String)*, определяющая состояние обработчика при первом запуске программного комплекса;
- *интерфейсы (Method Interfaces)*, необходимые для обмена информацией, являющейся результатом действий элемента маршрутизатора.

Модули Click не имеют встроенных буферов для построения очередей на входных и выходных портах. Вместо этого очереди в Click реализуются специальными Queue-обработчиками. Реализация механизма построения очередей даёт возможность прямого контроля над

параметрами маршрутизатора, а также возможность создания конфигураций, которые сложно реализовать другими методами.

5.4.2. Протоколы маршрутизации

Существует два основных способа определения маршрута и построения таблиц маршрутизации — *статический* и *динамический*.

Статическая маршрутизация

При использовании статического способа таблицы маршрутизации строятся администратором сети вручную. Для их построения используются специальные команды маршрутизатора (обычно это команда `route`, с помощью которой определяется маршрут для указанной сети). Параметрами этих команд служат адрес и маска сети назначения, адрес следующего маршрутизатора (next hop) для этой сети и имя или адрес интерфейса, через который должна быть передана дейтаграмма. Для корректной доставки дейтаграммы достаточно первых трёх параметров. При использовании внеклассовых сетей в таблице маршрутизации вполне могут появиться конфликтующие маршруты, поэтому указание маски является обязательным.

Построение полной таблицы маршрутизации, в которой были бы указаны все сети, образующие Интернет, невозможно из-за огромного количества этих сетей. Для того чтобы упростить процедуру построения таблицы маршрутизации, в неё может быть включён специальный узел, куда необходимо передать дейтаграммы, адрес сети назначения которых не указан в таблице маршрутизации. Этот специальный узел называется *шлюзом по-умолчанию (Default Gateway)* и применяется для маршрутизации в режиме «по-умолчанию». Для обозначения маршрута к Default Gateway в качестве адресов сети и маски принято использовать нулевые значения.

Основным недостатком статического метода является не размер и количество создаваемых вручную таблиц маршрутизации, а тот факт, что эти таблицы фиксированные и, следовательно, не могут реально соответствовать текущей конфигурации сети (нет возможности получения информации о новых сетях и нет выбора наиболее эффективного маршрута в сети).

Динамическая маршрутизация

При использовании динамической маршрутизации формирование маршрутных таблиц производится маршрутизаторами автоматически в результате постоянного выполнения специального алгоритма маршрутизации. В процессе его выполнения маршрутизатор передаёт своим соседям информацию об известных ему маршрутах, получая от

них взамен аналогичную информацию. После обработки полученной информации маршрутизатор строит заново или корректирует свою таблицу маршрутизации. Поскольку информация о состоянии маршрутов поступает на маршрутизатор постоянно, использование такого алгоритма обеспечивает постоянное соответствие содержимого таблицы маршрутизации реальному состоянию сети.

В зависимости от того, каким образом производится обмен маршрутной информацией между соседними маршрутизаторами, различают два типа алгоритмов маршрутизации:

- *алгоритмы вектора расстояния (Distance-Vector)*
  маршрутизатор через заранее определённые промежутки времени передаёт соседним маршрутизаторам содержимое своей таблицы маршрутизации;
- *алгоритмы состояния канала (Link-State)*
  маршрутизатор передаёт информацию только об изменениях состояния системы.

Во время построения маршрутной таблицы могут быть сформированы несколько маршрутов, ведущих в одну сеть. Для того чтобы маршрутизатор мог выбрать один из них в качестве предпочтительного, он должен использовать обобщённую характеристику качеств маршрута — *метрику (Metric)*.

Каждый алгоритм маршрутизации применяет свой алгоритм расчёта метрики. В наиболее простом случае в качестве метрики маршрута используется число узлов, отделяющих это маршрутизатор от сети назначения. Более сложные метрики учитывают характеристики физических каналов, составляющих маршрут. Некоторые алгоритмы маршрутизации для увеличения скорости информационного обмена позволяют одновременно использовать несколько маршрутов, ведущих к одной сети.

Совокупность сетей, находящихся под единым административным управлением, принято называть *автономной системой*.

Для определения внутренних маршрутов в автономных системах обычно используется один или несколько протоколов маршрутизации. В автономных системах этот класс протоколов принято называть *протоколами внутренней маршрутизации (Interior Gateway Protocol, IGP)*. Применение *протоколов внешней маршрутизации (Exterior Gateway Protocol, EGP)* позволяет администратору реализовать совокупность мер повышения надёжности и экономической эффективности информационного взаимодействия с внешними системами. В число параметров, используемых современным протоколом внешней маршрутизации для определения качества маршрута, входят предпочтительность маршрута, последовательность проходимых автономных систем и другие параметры.

Протокол RIP

Внутренний протокол маршрутизации *RIP (Routing Information Protocol)* использует алгоритм вектора расстояния для определения маршрута следования пакетов.

Функционирование маршрутизаторов по алгоритму вектора расстояния.

1. Маршрутизатор строит первичную таблицу маршрутизации, в которую помещает номера непосредственно подключённых сетей. Эта таблица содержит следующие поля:

   Address (Адрес) — адрес сети или узла назначения;
   Router (Маршрутизатор) — сетевой адрес первого маршрутизатора на маршруте к сети или узлу назначения;
   Interface (Интерфейс) — сетевой адрес или номер интерфейса связи с первым маршрутизатором;
   Metric (Метрика) — числовая характеристика маршрута от 0 до 15 (значение 0 соответствует непосредственно подключённой сети, метрика 15 указывает на недостижимость сети или узла назначения, в остальных случаях — соответствует количеству промежуточных маршрутизаторов на маршруте к сети или узлу назначения);
   Timer (Таймер) — показатель актуальности информации о сети или узле назначения (если информация не подтверждается источником в течение установленного временного интервала, запись о маршруте удаляется из таблицы).
2. Маршрутизатор рассылает оформленную в виде специального *сообщения об обновлении (Update)* текущую версию таблицы маршрутизации соседним маршрутизаторам.
3. При приёме аналогичного сообщения от соседнего маршрутизатора выполняются следующие действия:
   1) Если сообщение содержит информацию о сети, которой нет в таблице маршрутизации, адрес этой сети заносится в таблицу со следующими значениями полей:
      – *Router (Маршрутизатор)* — адрес источника сообщения;
      – *Interface (Интерфейс)* — адрес интерфейса, принявшего сообщение;
      – в поле *Metric (Метрика)* заносится значение соответствующего поля исходного сообщения, увеличенное на весовой коэффициент интерфейса (обычно все весовые коэффициенты интерфейсов принимаются равными 1);
      – значение поля *Timer (Таймер)* у созданной записи устанавливается равным утроенному периоду обновлений (90 с).

2) Если сообщение содержит информацию о сети, которая есть в таблице маршрутизации, выполняется сравнение содержимого полей Router существующей записи и принятого сообщения. Если источник маршрутной информации в обоих случаях был один и тот же, поле Metric существующей записи модифицируется по обычному алгоритму значением соответствующего поля принятого сообщения. Поле Timer для модифицированной записи формируется так же, как и для вновь созданной.
3) Если информацию об известной сети содержит сообщение, принятое от нового источника, маршрутизатор сравнивает содержимое полей Metric существующей записи и принятого сообщения. Если метрика существующего маршрута больше метрики нового маршрута, прежняя запись в таблице маршрутизации заменяется новой. В противном случае таблица маршрутизации никак не модифицируется.
4) В том случае, если значение поля Timer у существующей записи стало равным 0, запись удаляется из таблицы маршрутизации.

Процессы, описанные в двух последних пунктах, периодически повторяются, что позволяет динамически отслеживать изменения конфигурации сети.

В протоколе RIP в качестве предельного значения метрики маршрута используется значение 15. Сети, удалённые от данного узла на расстояние, которое превышает 15 переходов, считаются *недостижимыми (Unreachable)*.

Методы противодействия возникновению циклических маршрутов. Для противодействия возникновению циклических маршрутов алгоритмы маршрутизации Distance-Vector вообще и RIP в частности используют некоторые специальные методы:

– *Правило расщеплённого горизонта (Split Horizon).*
Информация о маршруте в некоторую сеть N, полученная от маршрутизатора , не может быть включена в регулярные обновления, отправляемые этому маршрутизатору. Использование этой процедуры позволяет гарантированно избежать появления циклических маршрутов между двумя соседними маршрутизаторами, повышает эффективность использования пропускной способности канала за счёт сокращения неинформативной составляющей сообщения об обновлении маршрутов. Однако в том случае, если циклический маршрут образован несколькими маршрутизаторами, применение этой процедуры не даст желаемого эффекта.
– *Правило отравленного обратного пути (Poison Reverse).*
Действует аналогично предыдущему правилу, однако, в отличие от процедуры расщеплённого горизонта, информация о маршруте в некоторую сеть N, полученная от маршрутизатора , включается в регулярные обновления, отправляемые этому маршрутизатору с

метрикой 16. В результате использования этой процедуры потенциально опасные маршруты будут удалены из таблицы маршрутизации. Но если при использовании чистой процедуры Split Horizon эти маршруты будут удалены по истечении определённого времени, то использование Poison Reverse приведёт к их мгновенному уничтожению.

– *Метод управляемых обновлений (Triggered Update).*
Наиболее мощным средством борьбы с длинными циклическими маршрутами является использование *апериодических управляемых обновлений маршрутов (Triggered Update)*. Маршрутизатор формирует обновление при каждом изменении своей таблицы маршрутизации, не дожидаясь наступления момента передачи очередного периодического обновления. При получении такого управляемого обновления последующий маршрутизатор скорректирует свою таблицу маршрутизации, а затем, в свою очередь, сформирует своё управляемое обновление, которое направит своим соседям. Таким образом, информация об изменении конфигурации распространяется по сети немедленно. Кроме того, вследствие особого дифференциального принципа формирования таких обновлений они распространяются по сети от источника только в нужных направлениях, поскольку маршрутизатор, не изменивший свою таблицу маршрутизации при получении управляемого обновления, не сформирует вторичное обновление и заблокирует его дальнейшее распространение.

Режимы RIP. При реализации RIP можно выделить следующие режимы:

– *инициализация* — посылается запрос для определения всех доступных интерфейсов;
– *получение таблиц маршрутизации* от других маршрутизаторов;
– *получен запрос* — посылается либо полная таблица маршрутизации, либо проводится индивидуальная обработка;
– *получен отклик* — проводится коррекция таблицы маршрутизации;
– *регулярные коррекции* — пересылка всей или части таблицы всем соседним маршрутизаторам каждые 30 с.

Формат сообщения RIP. Для взаимного обмена маршрутной информацией со своими соседями маршрутизаторы протокола RIP применяют сообщения специального формата (рис. 5.33).

Для отправки этих сообщений маршрутизаторы первой версии RIP обычно использовали широковещательный адрес (Broadcast) сетевого уровня. Особенно негативно эта особенность протокола проявлялась в сетях множественного доступа (например, Ethernet), где она могла приводить к значительному снижению эффективности использования сетевых ресурсов. В версии RIPv2 применяется специально выделенный

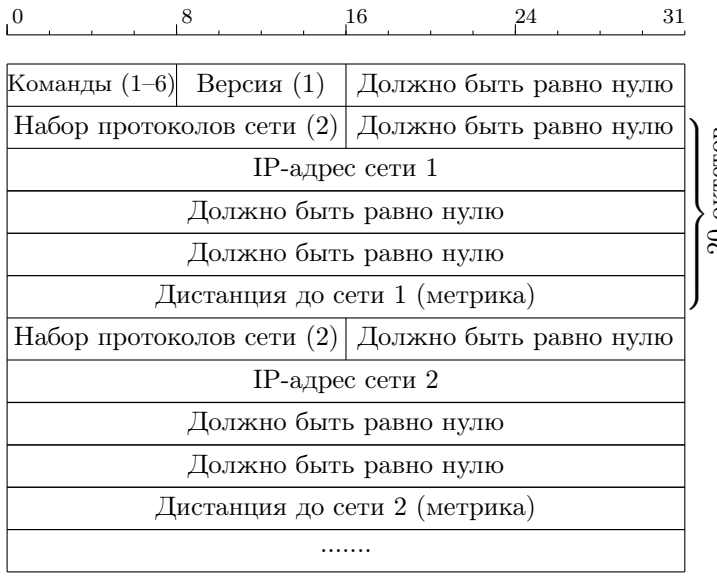


Рис. 5.33. Формат сообщения RIP

групповой адрес 224.0.0.9 или для передачи сообщения конкретному соседу — обычный одноадресный режим (Unicast).

Сообщения протокола RIP состоят из *заголовка* и следующих за ним *маршрутных записей (Route Entries, RTE)*. Обычно в сообщении протокола RIP содержится не более 25 маршрутных записей. То есть при передаче большой таблицы маршрутизатор должен использовать несколько последовательных сообщений.

Поле *Команда (Command)* может принимать следующие значения:

1 — запрос на получение частичной или полной маршрутной информации;

2 — отклик, содержащий информацию о расстояниях из маршрутной таблицы отправителя;

3 — включение режима трассировки;

4 — выключение режима трассировки;

5–6 — зарезервированы для внутренних целей SUN Microsystem.

Поле *Версия (Version)* указывает версию протокола RIP (1 или 2);

Поле *Набор протоколов сети (Address family identifier) i* ($i \leqslant 25$) указывает целое число шагов до данной сети (от 1 до 15).

Сообщения типа «запрос» используются для запроса на получение полной таблицы маршрутизации или её части. Обработка запроса

ведётся запись за записью (RTE за RTE). Для каждой маршрутной записи проверяется таблица маршрутизации на предмет того, есть ли там соответствующая запись. Если есть, то в поле маршрутной записи помещается метрика из таблицы маршрутизации. Если нет — в поле маршрутной записи помещается число 16, обозначающее бесконечную метрику. После того, как все маршрутные записи обработаны пакет отсылается обратно запрашивающему.

Сообщение типа «отклик» может быть ответом на конкретный запрос, регулярным сообщением обновления или сообщением обновления, вызванным изменением таблицы маршрутизации.

При получении сообщения типа «отклик» для каждого содержащегося в нём элемента вектора расстояний выполняются следующие действия:

- проверяется корректность указанных в сообщении адреса сети и маски;
- проверяется, не превышает ли метрика бесконечности:
  - некорректный элемент игнорируется;
  - если метрика меньше бесконечности, она увеличивается на 1;
- в таблице маршрутов производится поиск сети, указанной в рассматриваемом элементе вектора расстояний, причём если запись о такой сети в таблице маршрутов отсутствует и метрика в полученном элементе вектора меньше бесконечности, сеть вносится в таблицу маршрутов с указанной метрикой;
- в поле «Следующий маршрутизатор» заносится адрес маршрутизатора, приславшего сообщение;
- запускается таймер для принятой записи в таблице;
- если искомая запись присутствует в таблице с метрикой больше, чем объявленная в полученном векторе, в таблицу вносятся новые записи о метрике и, соответственно, об адресе следующего маршрутизатора и таймер для этой записи перезапускается;
- если искомая запись присутствует в таблице и отправителем полученного вектора был маршрутизатор, указанный в поле «Следующий маршрутизатор» этой записи, то таймер для этой записи перезапускается;
- более того, если при этом метрика в таблице отличается от метрики в полученном векторе расстояний, в таблицу вносится значение метрики из полученного вектора;
- во всех прочих случаях рассматриваемый элемент вектора расстояний игнорируется.

Недостатки RIP.

- Отсутствие поддержки спецификации CIDR.
  RIP-I воспринимает внеклассовые сети типа 10.1.0.0/16, 10.2.0.0/16 и т.д. как одну сеть класса A 10.0.0.0/8 и формирует для неё один

маршрут, что, естественно, приводит к потере пакетов, направляемых в указанные подсети. Этот недостаток был устранён во второй версии протокола путём введения в маршрутную информацию дополнительной характеристики *SUBNET MASK (маска сети назначения)*.
- Требует много времени для восстановления связи после сбоя в маршрутизаторе.
- Возможно возникновение циклов.
- Наличие лишь одного параметра определения маршрута — числа промежуточных маршрутизаторов.

Протокол RIPv2. RIPv2 является расширением протокола RIPv1. Он не внёс в протокол RIPv1 каких-либо серьёзных изменений в механизме или формате сообщения, а лишь добавил возможность передачи дополнительной информации. Изменения формата заголовка пакета RIPv2 коснулись лишь поля *Версия* и ранее неиспользуемых полей, содержащих теперь дополнительную информацию.

Так в новой версии протокола появилась возможность аутентификации передаваемых сообщений, для чего используется первая маршрутная запись в заголовке пакета[1].

Кроме того, стало возможным различать «внутренние» маршруты (полученные через RIP) от «внешних» (полученных от других протоколов маршрутизации, таких, как EGP, BGP).

Как было сказано ранее, в новой версии протокола RIP стало возможным при помощи поля *Маска подсети* различать не только сети, но и подсети.

В целях уменьшения использования полосы пропускания сетей протокол RIPv2 вместо адреса broadcast использует multicast-адрес — 224.0.0.9.

Протокол OSPF

Протокол *OSPF (Open Shortest Path First)* относится к протоколам маршрутизации на основе состояния канала (класс Link-State).

Функционирование маршрутизаторов по алгоритму состояния каналов. Как и все протоколы маршрутизации класса Link-State, протокол OSPF предназначен для построения внутренних маршрутов *автономной системы (Autonomous System)*.

Поскольку протокол OSPF обеспечивает иерархическую маршрутизацию, автономная система разбивается на независимые области по

[1] RFC 2453 специфицирует использование только одной схемы аутентификации — использование простого нешифруемого пароля.

функциональному принципу. Центральная область играет роль *магистрали (Backbone)* и используется для обеспечения информационного взаимодействия между остальными (периферийными) областями.

В зависимости от того, к какой области принадлежит маршрутизатор и какие информационные потоки через него проходят, различают четыре типа маршрутизаторов протокола OSPF:

- *внутренний маршрутизатор (Internal Router, IR)*;
- *пограничный маршрутизатор области (Area Border Router)*;
- *пограничный маршрутизатор автономной системы (AS Boundary Router, ASBR)*;
- *магистральный маршрутизатор (Backbone Router, BR)*.

Все маршрутизаторы OSPF принимают участие в формировании маршрутной информации автономной системы путём передачи специальных сообщений, содержащих информацию о текущем состоянии фрагмента сети. Эти сообщения называются *объявлением состояния канала (Link State Advertisement, LSA)*. Сообщения LSA обязательно формируются при любом изменении состояния контролируемого компонента сети. Для обеспечения большей надёжности сообщения LSA могут быть сформированы и при отсутствии каких-либо изменений в сети через достаточно большие интервалы времени, например, один раз за полчаса.

Принятые сообщения образуют в каждом маршрутизаторе *базу данных состояния сети (Link State Data Base)*. При получении сообщения об изменениях в структуре сети маршрутизатор вносит соответствующие изменения в свою копию базы данных. Таким образом, в каждый момент времени все базы данных маршрутизаторов, находящихся внутри одной автономной системы, являются идентичными и адекватно отображают структуру этой системы. Для того чтобы определить маршрут, по которому должна быть передана дейтаграмма, маршрутизатор на основании своей копии базы данных строит дерево кратчайших путей, в вершине которого размещает самого себя (используя алгоритм Дейкстры). Построение кратчайших путей маршрутизатор выполняет всякий раз, когда происходит изменение состояния сети.

Существенной особенностью протокола маршрутизации OSPF является специальная процедура информационного обмена между маршрутизаторами в сетях с множественным доступом (например, Ethernet). Маршрутизаторы, подключённые к одной и той же сети, называются *соседними маршрутизаторами (Neighboring Routers)*. Маршрутизаторы протокола OSPF устанавливают и обслуживают соседские отношения, используя специальный дополнительный протокол *Hello*. С помощью этого протокола определяется состав подключённых к сети маршрутизаторов, их работоспособность и производится выбор одного из них в качестве *назначенного маршрутизатора (Designated Router, DR)*. Назначенный маршрутизатор выбирается для того, чтобы ис-

ключить возможность многократного представления информации об одной сети. Он формирует сообщения, содержащие список подключённых к сети маршрутизаторов, и передаёт содержимое текущей базы данных по запросу, полученному от одного из них. Если по каким-либо причинам назначенный маршрутизатор перестал функционировать, его функции автоматически переходят к *запасному назначенному маршрутизатору (Backup Designated Router, BDR)*, выбираемому одновременно с основным.

Для передачи маршрутной информации маршрутизаторы протокола OSPF используют различные типы обновлений о состоянии сетевых компонентов (LSA). Процесс распространения LSA в пределах автономной системы называется *затоплением (Flooding)*.

Для хранения маршрутной информации протокола OSPF маршрутизаторы используют специальные *топологические базы данных (Link-State Database)*. База данных формируется из принятых сообщений LSA и отображает текущее состояние и структуру информационных связей в рассматриваемой области маршрутизации. На основании этой базы каждый маршрутизатор строит дерево кратчайших путей, соединяющих его самого с остальными компонентами области, и собственно таблицу маршрутизации.

Формат сообщений протокола OSPF. Формат заголовка сообщений протокола OSPF приведён на рис. 5.34.

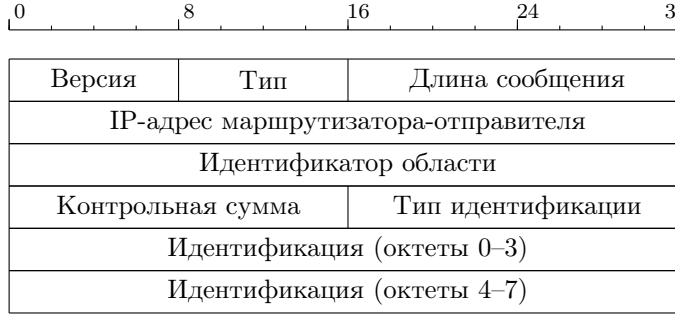


Рис. 5.34. Формат заголовка сообщений протокола OSPF

Поле *Версия (Version No.)* указывает версию протокола (=2).

Поле *Тип (Packet Type)* идентифицирует функцию сообщения и может принимать следующие значения:

1 — сообщение является сообщением Hellow (используется для проверки доступности маршрутизатора);

2 — сообщение является описанием базы данных;

3 — сообщение является запросом состояния канала;
4 — сообщение информирует об изменении состояния канала;
5 — сообщение является подтверждением получения сообщения о статусе канала.

Поле *Длина пакета (Packet Length)* определяет длину блока (включая заголовок) в октетах.

Поле *Идентификатор области (Area ID)* представляет собой 32-битный код, идентифицирующий область, которой принадлежит данный пакет.

Поле *Контрольная сумма (Checksum)* содержит контрольную сумму IP-пакета, включая поле *Тип идентификации*. Контрольное суммирование производится по модулю 1.

Поле *Тип идентификации (AU type)* имеет значение 0, если отсутствует контроль доступа, и 1 — в противном случае .

Формат сообщения Hellow протокола OSPF приведён на рис. 5.35.

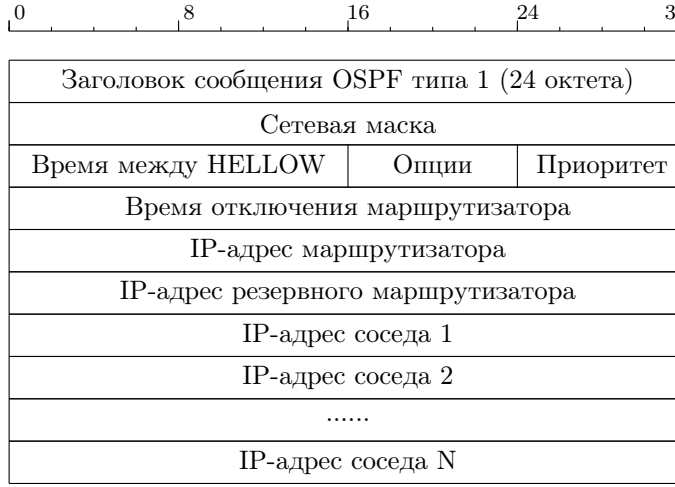


Рис. 5.35. Формат сообщения Hellow протокола OSPF

Поле *Сетевая маска* соответствует маске подсети интерфейса.

Поле *Время между сообщениями HELLOW* содержит значение времени в секундах между сообщениями Hellow.

Поле *Приоритет* определяет уровень приоритета маршрутизатора.

Поле *Время отключения маршрутизатора* определяет временно́й интервал в секундах, по истечении которого не отвечающий маршрутизатор считается вышедшим из строя.

Поля *IP-адрес маршрутизатора* и *IP-адрес резервного маршрутизатора* указывают, куда надо послать сообщение.

Поля *IP-адрес соседа i* образуют список адресов соседних маршрутизаторов, от которых недавно были получены сообщения Hellow.

Поле *Опции* (8 бит) информирует о состояние канала и описывает базу данных. Его формат приведён на рис. 5.36.

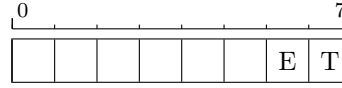


Рис. 5.36. Формат поля *Опции* протокола OSPF с типом сообщения Hellow

Бит E характеризует возможность внешней маршрутизации и имеет значение только в сообщениях типа Hellow, в остальных сообщениях данный бит должен быть обнулён (т.е. маршрутизатор не будет посылать или принимать маршрутную информацию от внешних автономных систем). Бит T определяет сервисные возможности маршрутизатора (Type of Service, ToS). Если $T = 0$, то маршрутизатор поддерживает лишь один вид услуг.

Формат сообщения OSPF о маршрутах приведён на рис. 5.37.

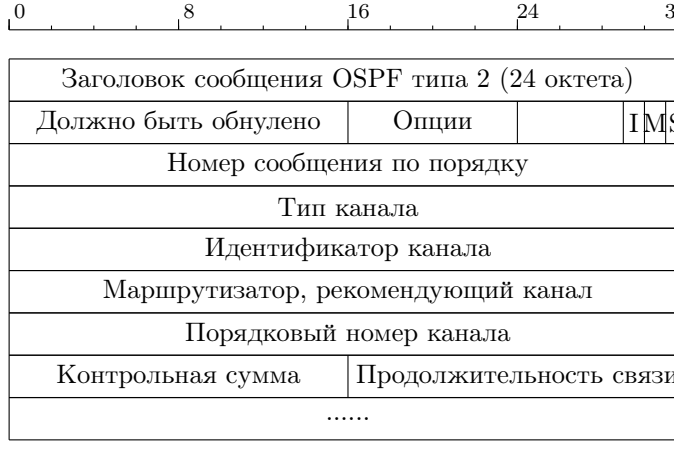


Рис. 5.37. Формат OSPF-сообщения о маршрутах

Поля, начиная с поля *Тип канала*, повторяются для каждого описания канала.

Содержимое базы может пересылаться по частям. В стартовом сообщении бит I устанавливается в 1, в сообщении-продолжении бит M устанавливается в 1. Бит S определяет, послано сообщение сервером ($S = 1$) или клиентом ($S = 0$).

Поле *Номер сообщения по порядку* служит для контроля пропущенных блоков.

Поле *Тип канала* характеризует объявление о маршруте и может принимать следующие значения:

1 — описание каналов маршрутизатора (состояние его интерфейсов);

2 — описание сетевых каналов (перечень маршрутизаторов, непосредственно связанных с сетью);

3 или 4 — сводное описание каналов, в которое входят маршруты между отдельными областями сети (тип 3 приписан маршрутам, ведущим к сетям, а тип 4 — маршрутам, ведущим от сетей);

5 — описание внешних связей автономной системы.

Поле *Идентификатор канала* определяет характер канала (идентификатором может быть IP-адрес маршрутизатора или сети).

Поле *Маршрутизатор, рекомендующий канал* определяет адрес этого маршрутизатора.

Поле *Порядковый номер канала* позволяет маршрутизатору контролировать порядок прихода сообщений и их потерю.

Поле *Продолжительность связи* определяет время в секундах с момента установления связи.

Формат OSPF-запроса маршрутной информации приведён на рис. 5.38.

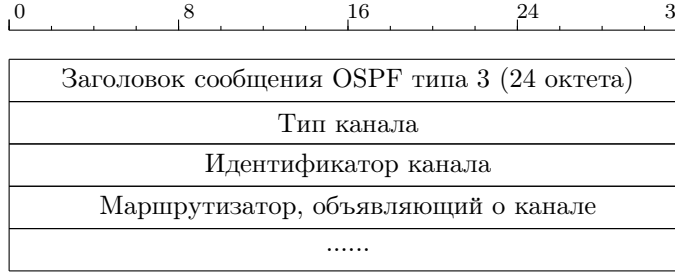


Рис. 5.38. Формат OSPF-запроса маршрутной информации

Формат сообщения о получении OSPF-пакета приведён на рис. 5.39.

Формат OSPF-сообщения об изменении маршрутов приведён на рис. 5.40.

Причины посылки сообщений об изменении маршрута:

1. продолжительность связи достигла предельного значения;
2. изменилось состояние интерфейса;

Рис. 5.39. Формат сообщения о получении OSPF-пакета

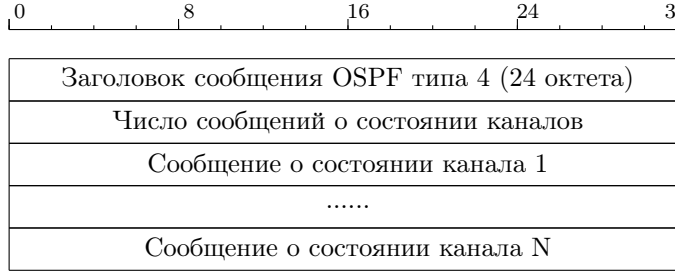


Рис. 5.40. Формат OSPF-сообщения об изменении маршрутов

3. произошли изменения в маршрутизаторе сети;
4. произошли изменения в одном из соседних маршрутизаторов;
5. изменилось состояние одного из внутренних маршрутов;
6. изменение состояния межзонного маршрута;
7. появление нового маршрутизатора, подключённого к сети;
8. изменение виртуального маршрута одним из маршрутизаторов сети;
9. изменение одного из внешних маршрутов;
10. маршрутизатор перестал быть пограничным для автономной системы.

Достоинства протокола OSPF. В отличие от универсальных протоколов (например, RIP), протокол OSPF предназначен для построения маршрутов только в сетях TCP/IP.

Основными достоинствами протокола OSPF являются:

- отсутствие ограничений на размер сети;
- поддержка внеклассовых сетей;
- передача сообщений протокола с использованием multicast-адресов, причём отдельные адреса используются для передачи и приёма информации о маршрутах в информационной системе;
- высокая скорость установления маршрутов при изменении состояния системы;

– встроенная процедура установления подлинности источника маршрутной информации;
– возможность использования нескольких параллельных путей к одному пункту назначения (Load Balancing);
– композитная метрика;
– иерархическая маршрутизация.

Протокол BGP

Протокол пограничного шлюза (Border Gateway Protocol,BGP) является протоколом маршрутизации между автономными системами. Данный протокол работает поверх протокола транспортного уровня. Это позволяет не нагружать сервисы обработки протокола BGP механизмами фрагментации или обеспечения достоверности доставки пакетов. Схемы аутентификации протоколов транспортного уровня также могут быть использованы BGP в дополнение к собственной системе аутентификации. Кроме того, хотя BGP разработан как протокол маршрутизации между автономными системами, он может использоваться для маршрутизации и внутри автономных систем.

Основным предназначением BGP является обеспечение обмена информацией с другими BGP-системами о досягаемости определённых сетей или хостов. Эта информация должна содержать набор маршрутов к данной сети, т.е. должны быть указаны все промежуточные автономные системы. Такой информации вполне достаточно для того, чтобы построить граф соединений между автономными системами и проконтролировать возможные маршрутные петли. На основании этих данных BGP выбирает оптимальный маршрут и передаёт эту информацию своим соседям.

Отличия протокола BGP от других протоколов маршрутизации.

Протокол BGP нельзя отнести ни к классу дистанционно-векторных, ни к классу протоколов маршрутизации на основе состояния канала. Ниже приведены характерные отличия протокола BGP от других протоколов маршрутизации.

– *Коммуникация между автономными системами.*
Поскольку протокол BGP относится к протоколам внешнего шлюза, его основное назначение — обеспечить обмен информацией между двумя автономными системами.
– *Координирование работы нескольких спикеров BGP.*
Если в состав автономной системы входит несколько маршрутизаторов, каждый из которых обменивается информацией с равным ему по рангу маршрутизатором внешней автономной системы (их называют *спикерами BGP*), протокол BGP может использоваться

для координации работы всего набора маршрутизаторов. Это гарантирует, что маршрутизаторы распространяют непротиворечивую информацию.

– *Распространение информации о достижимости.*
Протокол BGP позволяет автономной системе сообщить информацию о расположенных в ней получателях, а также о тех получателях, доступ к которым осуществляется через данную автономную систему. Кроме того, с помощью протокола BGP подобную информацию можно получить от других автономных систем.
– *Принцип ближайшего перехода.*
Подобно дистанционно-векторным протоколам маршрутизации, протокол BGP предоставляет информацию об адресе *ближайшей точки перехода* для каждого получателя.
– *Поддержка различной политики маршрутизации.*
В отличие от многих дистанционно-векторных протоколов, которые сообщают только ту маршрутную информацию, которая находится в локальной таблице маршрутизации, протокол BGP обеспечивает политику маршрутизации в зависимости от выбора администратора. В частности, маршрутизатор, работающий под управлением протокола BGP, можно настроить так, чтобы он различал получателей, доступ к которым осуществляется через компьютеры его автономной системы, и получателей, анонсированных другими автономными системами.
– *Надёжный транспортный протокол.*
Протокол BGP отличается от других протоколов, передающих информацию о маршрутизации, тем, что он предполагает использование надёжного транспортного протокола. Таким образом, для обмена информацией в протоколе BGP используется исключительно транспортный протокол TCP.
– *Информация о маршруте.*
Кроме указания списка возможных получателей и адреса ближайшей точки перехода для каждого из них в сообщении протокола BGP анонсируется также маршрутная информация, которая позволяет узнать, через какие автономные системы проложен маршрут к конкретному получателю.
– *Передача обновлений.*
Чтобы не создавать дополнительную нагрузку на сеть, в каждом сообщении протокола BGP об обновлении не передаётся полная маршрутная информация. Вместо этого обмен полной информацией происходит только один раз, а в следующих сообщениях передаются только изменения.
– *Поддержка бесклассовой адресации.*
Протокол BGP поддерживает CIDR-адреса. Это означает, что программа протокола BGP не полагается на методы идентификации IP-адресов, а вместе с каждым адресом отсылает и его маску.
– *Объединение маршрутов.*

Чтобы не создавать дополнительной нагрузки на сеть, протокол BGP позволяет отправителю накапливать информацию о маршрутах и отсылать в одном пакете данные сразу о нескольких, связанных между собой получателях.

– *Аутентификация.*
Протокол BGP позволяет получателю удостоверить подлинность сообщений (т.е. подтвердить «личность» отправителя).

Функции протокола BGP и виды сообщений. В процессе взаимодействия по протоколу BGP выполняется три основных действия:

1. получение согласия сторон на взаимодействие по протоколу BGP и аутентификацию (при этом два равноправных маршрутизатора устанавливают соединение по протоколу TCP и обмениваются сообщениями, которые подтверждают, что обе стороны согласны вступить в процесс обмена информацией);
2. каждая сторона отсылает информацию о доступности или недоступности получателей (это означает, что отправитель может сообщить о возможности доступа к одной или нескольким сетям получателя (при этом указывается адрес ближайшей точки перехода для каждой сети) или, напротив, может заявить, что одна или несколько сетей, о которых сообщалось ранее, более недоступны);
3. осуществление постоянного контроля над правильностью функционирования взаимодействующих пар маршрутизаторов и сетевых соединений.

В протоколе BGP определено четыре основных типа сообщений: OPEN (инициализирует процесс), UPDATE (аннулирует маршрутную информацию), NOTIFICATION (отвечает на неверное сообщение, KEEPALIVE (выполняет активную проверку возможности соединения между BGP-парами).

В начале каждого сообщения протокола BGP расположен заголовок фиксированного формата, с помощью которого определяется тип сообщения (рис. 5.41).

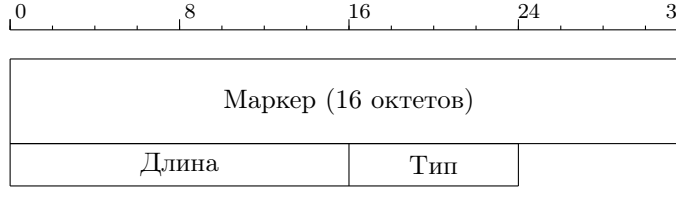


Рис. 5.41. Формат сообщения BGP

В поле *Маркер (Marker)* (16 октетов) заносится значение, которое обе стороны «договорились» использовать в качестве метки начала сообщения.

В поле *Длина (Length)* (2 октета) указывается общая длина сообщения в октетах. Минимальный размер сообщения составляет 19 октетов (для типа сообщения, в котором после заголовка нет данных). Максимально допустимая длина сообщения составляет 4096 октетов.

Наличие поля маркера является нехарактерным для сетевых протоколов. В исходном сообщении маркер состоит из всех единиц. Если взаимодействующие между собой маршрутизаторы «договорятся» об использовании механизма аутентификации, в поле маркера может содержаться информация об аутентификации. В любом случае обе стороны должны согласовать, какое значение будет внесено в это поле, чтобы его можно было в дальнейшем использовать для выполнения синхронизации.

Обмен всеми типами сообщений в протоколе BGP происходит через протокол TCP, в котором невозможно определить, где заканчивается одно сообщение и начинается другое. В такой среде ошибка, произошедшая на стороне одного из участников соединения, может привести к потере пакета, а получатель никогда не узнает об ошибке. Таким образом, чтобы обеспечить синхронные действия отправителя и получателя, BGP помещает в начало каждого сообщения некоторую известную обеим сторонам последовательность октетов, и перед дальнейшей обработкой сообщения требует от получателя подтвердить, что данное значение не повреждено.

BGP-сообщения OPEN (рис. 5.42) является запросом BGP-соединения и передаётся для организации сеанса связи между равноправными BGP-маршрутизаторами. Принявший это сообщение маршрутизатор подтверждает установление соединения, передавая сообщение KEEPALIVE (сообщение KEEPALIVE содержит только заголовок и обеспечивает сброс таймера удержания соединения).

Рис. 5.42. Формат BGP-сообщения OPEN

Поле *Версия* (длина 8 бит) указывает на версию протокола BGP.

Поле *Моя автономная система* (длина 16 бит) содержит идентификатор автономной системы отправившего сообщение маршрутизатора.

Поле *Время сохранения (Hold Time)* (длина 16 бит) указывает на максимальное время (в секундах) между приходами сообщений KEEPALIVE, используемых для монитронига активности соединения.

Поле *BGP-идентификатор* (длина 32 бита) содержит идентификатор маршрутизатора (один из адресов интерфейсов).

Поле *Код идентификации* (длина 8 бит) содержит длину поля *Идентификационные данные*, содержащее различные опции.

Сообщение UPDATE рассылается маршрутизатором BGP с целью внесения изменений в таблицы маршрутизации. Формат BGP-сообщения UPDATE приведён на рис. 5.43.

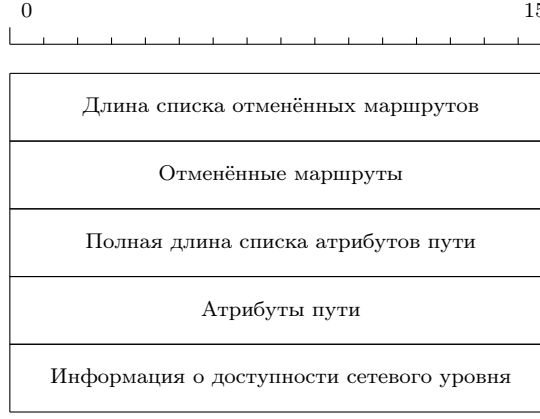


Рис. 5.43. Формат BGP-сообщения Update

Сообщение UPDATE состоит из трех частей переменной длины: списка отменённых (недействительных) маршрутов, списка атрибутов пути и списка сетей, к которым эти атрибуты относятся. Две последние части представляют собой собственно информацию о маршруте в указанные сети.

Список адресов сетей и список недействительных маршрутов представляют собой списки элементов, состоящих из длины префикса и собственно сетевого адреса.

Сообщениями-уведомлениями (NOTIFICATION) BGP-маршрутизаторы обмениваются при возникновении ошибок. Такие сообщения содержат в себе код ошибки (например, ошибка заголовка, ошибка в сообщении OPEN, ошибка в сообщении UPDATE и т.д.).

5.5. Коммутация пакетов по меткам (MPLS)

Технология коммутации пакетов по меткам в многопротокольных сетях (Multiprotocol Label Switching, MPLS) представляет собой механизм передачи данных, который эмулирует различные свойства сетей с коммутацией каналов поверх сетей с коммутацией пакетов (RFC 3031[1]).

В традиционной IP-сети при передаче пакетов маршрутизаторы на основе данных заголовков (адрес назначения) принимают решение о выборе дальнейшего маршрута.

В сетях на базе протокола MPLS заголовки передаваемых пакетов не анализируются при прохождении через маршрутизаторы, а переадресация осуществляется исключительно на основе меток.

5.5.1. Архитектура MPLS

В основе архитектуры MPLS, как следует из названия, лежит процесс коммутации пакетов по меткам. *Метка (Label)* представляет собой короткий идентификатор фиксированной длины, который определяет принадлежность пакета к некоторому классу на каждом из участков коммутируемого маршрута.

Сеть MPLS делится на две области — *ядро* и *граничную область* (рис. 5.44).

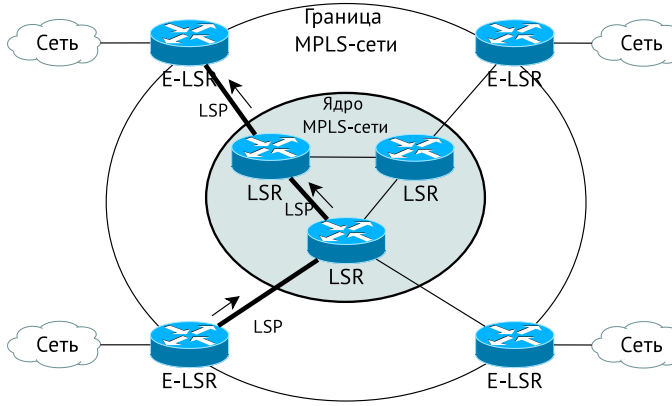


Рис. 5.44. Архитектура сети MPLS

[1] *Rosen E.*, *Viswanathan A.*, *Callon R.* Multiprotocol Label Switching Architecture, RFC 3031. URL: `http://www.ietf.org/rfc/rfc3031.txt`.

Ядро образуют устройства *Label-Switch Routers (LSR)* — маршрутизаторы, поддерживающие как обычную IP-маршрутизацию, так и коммутацию по меткам. Маршрутизаторы ядра отвечают только за коммутацию. Границу сети MPLS образуют *граничные маршрутизаторы (Edge LSR, E-LSR)*, осуществляющие классификацию поступающих в MPLS-сеть пакетов, их фильтрацию, управление трафиком и т.п. Первая метка, устанавливаемая на граничном маршрутизаторе, определяет *маршрут следования (Label Switch Path, LSP)* пакета через MPLS-домен.

Множество подсетей, поставленное в соответствие конкретному LSP, образуют *класс эквивалентности (Forwarding Equivalence Classes, FEC)*. Каждый из классов FEC обрабатывается отдельно — строится свой путь LSP, выделяется своя ширина полосы пропускания канала и т.п.

LSR выполняет две функции — *маршрутизацию* и *коммутацию по меткам*.

Процесс маршрутизации функционирует на базе внутреннего протокола маршрутизации (например, OSPF). LSR получает маршрутную информацию от соседних маршрутизаторов и формирует таблицу маршрутизации, которая используется для маршрутизации IP-пакетов.

Процесс коммутации функционирует на базе *протокола обмена метками (Label Distribution Protocol, LDP)*, ставящего в соответствие конкретному значению метки определённый маршрут LSP.

5.5.2. Формат MPLS-метки

На рис. 5.45 представлен формат MPLS-метки (RFC 3032[1]).

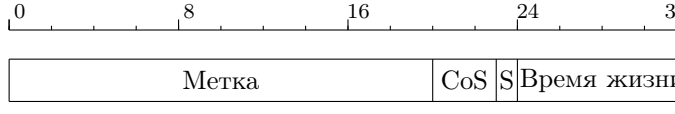


Рис. 5.45. Формат MPLS-метки

Поле *Метка (Label)* (длина 20 бит) содержит код метки, по которой осуществляется коммутация.

Зарезервированные значения меток:

– 0 (IPv4 Explicit NULL Label) — указывает, что стек меток должен быть очищен, а переадресация пакета должна основываться на заголовке IPv4;

[1]MPLS Label Stack Encoding, RFC 3032 / E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta. URL: `http://www.ietf.org/rfc/rfc3032.txt`.

– 1 (Router Alert Label) — указывает на то, что переадресация пакета определяется меткой;
– 2 (IPv6 Explicit NULL Label) — указывает, что стек меток должен быть очищен, а переадресация пакета должна основываться на заголовке IPv6;
– 3 (Implicit NULL Label) — значение, присваиваемое маршрутизатором.

Поле *(Class of Service, CoS)* (длина 3 бита) характеризует класс обслуживания пакета.

Поле *S* может принимать значение 0 или 1, указывая, является ли метка последней в стеке меток, присвоенных одному пакету[1].

Поле *Время жизни (Time-to-Live, TTL)* (длина 8 бит) указывает в общем случае число возможных промежуточных узлов.

MPLS-метка передаётся в составе любого пакета, причём способ её присоединения к пакету зависит от используемой технологии канального уровня. MPLS-метка добавляется между заголовком кадра (второй уровень ISO/OSI) и заголовком пакета (третий уровень модели ISO/OSI) (рис. 5.46).

Рис. 5.46. Расположение MPLS-метки

5.5.3. Label Distribution Protocol

Протокол распространения меток (Label Distribution Protocol, LDP) предназначен для построения целостных маршрутов LSP (RFC 3036[2]). LDP представляет собой набор процедур и сообщений, с помощью которых LSR формирует сетевой маршрут LSP путём установления соответствия между маршрутной информацией и каналами передачи данных.

В функции LDP входит: определение соседнего маршрутизатора, управление сессией, рассылка меток, уведомление об ошибках.

Обмены сообщениями LDP осуществляются путём посылки протокольных данных LDP (PDU) через LDP-секцию TCP-соединений. При этом каждый LDP PDU может содержать более одного LDP-сообщения.

[1] В рамках архитектуры MPLS вместе с пакетом разрешено передавать не одну метку, а целый стек.

[2] LDP Specification, RFC 3036 / L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas. URL: `http://www.ietf.org/rfc/rfc3036.txt`.

Каждый LDP PDU представляет собой LDP-заголовок (рис. 5.47), за которым следует одно или более LDP-сообщений.

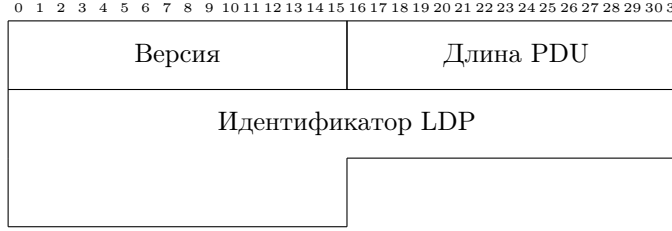


Рис. 5.47. Заголовок LDP

Поле *Версия (Version)* (длина 2 байта) содержит код номера версии протокола.

Поле *Длина PDU (PDU Length)* (длина 2 байта) указывает общую длину PDU в октетах, исключая поля версии и длины PDU.

Поле *Идентификатор LDP (LDP Identifier)* (длина 6 байт) однозначно идентифицирует пространство меток LSR-отправителя. При этом первые четыре октета идентифицируют LSR и должны быть уникальными, а последние два октета идентифицируют пространство меток заданного LSR.

Все сообщения LDP имеют определённый формат (рис. 5.48).

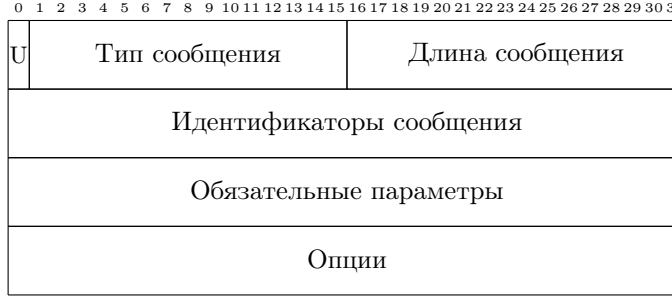


Рис. 5.48. Формат LDP-сообщений

Поле U представляет собой бит неизвестного сообщения; при $U = 1$ сообщение игнорируется.

Поле *Тип сообщения (Message Type)* идентифицирует тип сообщения.

Поле *Длина сообщения (Message Length)* указывает суммарную

длину в октетах полей идентификатора сообщения, обязательных параметров и опций.

Поле *Идентификатор сообщения (Message ID)* идентифицирует сообщение.

Поле *Обязательные параметры* представляет собой набор необходимых параметров.

Поле *Опции* представляет собой набор необязательных параметров.

В LDP определены следующие типы сообщений:

- *Hello* — определение соседнего маршрутизатора;
- *инициализация (Init)* — процедура установления сессии;
- *KeepAlive* — используется для поддержания активного статуса LDP-сессии;
- *адрес (Address Message)* — анонсирование адреса интерфейса маршрутизатора;
- *отзыв адреса (Address Withdraw)* — отзыв ранее анонсированного адреса интерфейса;
- *присвоение метки (Label Mapping)* — сообщение о присвоении метки;
- *запрос метки (Label Request)* — запрос метки у соседнего маршрутизатора с целью установления соответствия значения метки и FEC;
- *запрос ликвидации метки (Label Release)* — подтверждение получения метки в сообщении Label Mapping;
- *отзыв метки (Label Abort Request)* — сигнал соседнему маршрутизатору о невозможности продолжения использования ассоциации FEC–метка;
- *освобождение метки (Label Withdraw)* — сообщение о ненужности ранее полученной метки.

Установление LDP сессии происходит по следующему сценарию:

- при помощи обмена сообщениями *Hello* соседние маршрутизаторы определяют транспортные адреса друг друга;
- один из маршрутизаторов становится активным;
- активный маршрутизатор устанавливает TCP/IP сессию на порт 646 и посылает сообщение *Init*, включающее в себя параметры LDP-сессии;
- пассивный маршрутизатор проверяет полученные параметры LDP-сессии на совместимость с локальными настройками LDP и посылает ответное сообщение *Init* со своими параметрами LDP-сессии;
- активный маршрутизатор также проверяет полученные параметры LDP-сессии на совместимость с локальными настройками LDP, после чего сессия считается установленной.

Если на каком-то этапе возникают ошибки, то сессия считается неустановленной, а маршрутизатор, обнаруживший ошибку, посылает сообщение *Shutdown* или *Reject* своему соседу.

LDP-сессия будет установлена, если совпадают версии протокола LDP и совпадают режимы распространения информации о метках.

5.5.4. Сервисы на базе MPLS

На базе MPLS возможна организация следующих сервисов:
- MPLS/VPN — создание распределённых виртуальных частных сетей (Virtual Private Network, VPN) на крупных сетях без организации туннелей и шифрования;
- MPLS/TrafficEnginering — гибкое управление потоками трафика внутри MPLS-домена и более полное использование канальной инфраструктуры сети;
- AnyTransportOverMPLS — прозрачная передача через MPLS-домен кадров ATM, Frame Relay, Ethernet и т.п.

5.5.5. Особенности MPLS

Главной особенностью MPLS является отделение процесса коммутации пакета от анализа IP-адресов в его заголовке. Вся информация о маршруте содержится в метке, и пакету не требуется нести адреса промежуточных узлов, что улучшает управление распределением нагрузки в сети.

В сетях MPLS есть возможность организации при помощи протокола RSVP явной коммутации пакетов через так называемые туннели, что повышает эффективность загрузки каналов в MPLS-сети с альтернативными путями, поскольку трафик с определённой меткой идёт по конкретному пути с заданными параметрами качества обслуживания. Такое решение снимает необходимость иметь маршрутную информацию на всех маршрутизаторах в сети оператора.

Ещё одной важной особенностью сетей MPLS является возможность разделения IP-трафика и создания VPN-соединений между различными узлами, а также независимость адресных пространств операторской и клиентских сетей. Такое решение даёт возможность масштабирования сети, интеграции сети с другими сервисами IP.

5.5.6. Вопросы по разделу

1. Укажите функции и услуги сетевого уровня модели ISO/OSI.
2. Укажите основные отличия между протоколами IPv4 и IPv6.
3. Опишите схему адресации IPv4. Приведите классификацию адресов.
4. Опишите схему адресации IPv6. Приведите классификацию адресов.
5. В чем заключается назначение и функциональность протокола ICMP?
6. Опишите процедуру преобразования адресов с помощью протоколов ARP и RARP.
7. Приведите классификацию протоколов маршрутизации.
8. Опишите схему работы алгоритма вектора расстояния и схему работы алгоритма состояния канала.

9. Опишите схему функционирования протокола RIP. Укажите основные отличия между протоколами RIPv1 и RIPv2.
10. Опишите схему функционирования протокола OSPF.
11. В чем отличия протокола BGP от других протоколов динамической маршрутизации?
12. Опишите схему функционирования протокола BGP.
13. Какие элементы составляют архитектуру сети MPLS и какие функции они выполняют?
14. В чем заключается функциональность протокола LDP? Опишите схему его работы.

5.6. Примеры заданий

Задание 5.1 — Разбейте сеть 100.0.0.0/8 на 5 подсетей.

Ответ (Задание 5.1) — Дана сеть 100.0.0.0/8. Длина сетевого префикса — 8. Сети соответствует маска 255.0.0.0 и broadcast-адрес 100.255.255.255/8. Данную сеть можно разбить на $2^8 = 256$ подсетей.

Разобьём сеть сначала на 4 подсети. Для этого под идентификатор сети надо выделить дополнительно 2 бита, чтобы получить 4 различные комбинации: `00`, `01`, `10`, `11`. Эти комбинации бит определяют вид второго октета адреса: `00000000` даёт число 0 в десятичной форме записи, `01000000` — число 64, `10000000` — число 128, `11000000` — число 192.

Сетевой префикс будет иметь длину 10, а маска будет иметь вид 255.192.0.0 (или в двоичной форме `11111111 11000000 00000000 00000000`).

Чтобы получить broadcast-адрес, надо зафиксировать сетевую часть адреса, а биты, относящиеся к номеру хоста, положить равными 1. Так сеть 100.64.0.0/10 в двоичной форме запишется как

`01100100 01000000 00000000 00000000.`

Её broadcast-адрес в двоичной форме будет иметь вид

`01100100 01111111 11111111 11111111,`

а в десятичной форме — 100.127.255.255.

Таким образом, получим 4 подсети.

адрес подсети	broadcast-адрес	маска
100.0.0.0/10	100.63.255.255/10	255.192.0.0
100.64.0.0/10	100.127.255.255/10	255.192.0.0
100.128.0.0/10	100.191.255.255/10	255.192.0.0
100.192.0.0/10	100.255.255.255/10	255.192.0.0

Одну из подсетей, например 100.64.0.0/10, разобьём ещё на 2 подсети. Для этого под идентификатор сети надо выделить дополнительно ещё 1 бит. Маска будет иметь вид 255.224.0.0. Тогда получим ещё 2 подсети:

адрес подсети	broadcast-адрес	маска
100.64.0.0/11	100.81.255.255/11	255.224.0.0
100.96.0.0/11	100.127.255.255/11	255.224.0.0

5.7. Задания для самостоятельной работы

Задание 5.2 — Разделите сеть 10.128.0.0/9 на 13 подсетей.

Задание 5.3 — Разделите сеть 192.168.0.0/24 на 9 подсетей.

Задание 5.4 — Разделите сеть 172.16.0.0/11 на 7 подсетей.

Глава 6. Транспортный уровень

В главе рассмотрены основные протоколы транспортного уровня стека TCP/IP: UDP, TCP, SCTP, DCCP. В основу раздела легли материалы из источников [1; 2], а также интернет-стандарты[1].

В результате освоения данной темы студент должен:

знать:

– структуры заголовков протоколов транспортного уровня стека TCP/IP;
– принципы передачи данных на транспортном уровне;

уметь:

– определять область применения того или иного транспортного протокола;

владеть:

– способностью анализировать данные заголовков протоколов транспортного уровня.

6.1. Основная концепция протоколов транспортного уровня

На транспортном уровне организована служба надёжной доставки данных для верхних уровней, использующая управление потоком и коррекцию ошибок в сквозном потоке. Некоторые реализации транспортного уровня дополнительно осуществляют сегментацию данных при их отправке и воссоздании на приёмной стороне. Кроме того, транспортный уровень предоставляет приложению одно или несколько виртуальных соединений, связывающих оконечные точки.

[1] *Postel J.* User Datagram Protocol, RFC 768. URL: `http://www.faqs.org/rfcs/rfc768.html`; *Postel J.* Transmission Control Protocol, RFC 793. URL: `http://www.faqs.org/rfcs/rfc793.html`; Stream Control Transmission Protocol, RFC 2960 / R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. URL: `http://www.faqs.org/rfcs/rfc2960.html`; *Ong L.*, *Yoakum J.* An Introduction to the Stream Control Transmission Protocol (SCTP), RFC 3286. URL: `http://www.faqs.org/rfcs/rfc3286.html`; *Mogul J.*, *Deering S.* Path MTU discovery, RFC 1191. URL: `http://www.faqs.org/rfcs/rfc1191.html`; *Floyd S.*, *Handley M.*, *Kohler E.* Problem Statement for the Datagram Congestion Control Protocol (DCCP), RFC 4336. URL: `http://tools.ietf.org/html/rfc4336`; *Kohler E.*, *Handley M.*, *Floyd S.* Datagram Congestion Control Protocol (DCCP), RFC 4340. URL: `http://tools.ietf.org/html/rfc4340`; *Ramakrishnan K.*, *Floyd S.*, *Black D.* The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168. URL: `http://tools.ietf.org/html/rfc3168`.

Сегментация данных позволяет разделить большой блок данных, переданных приложением, на более мелкие фрагменты, которые способен передать сетевой уровень. На сетевом уровне выполняется инкапсуляция заголовков пакетов транспортного протокола и прикладных данных, а сформированный пакет передаётся на канальный уровень.

Управление потоком на транспортном уровне обычно сопровождается ограничением числа пакетов, которые могут быть посланы без подтверждения их приёма.

На транспортном уровне семейства протоколов TCP/IP применяются два основных протокола — ориентированный на соединение протокол TCP и не требующий соединения протокол UDP.

Важной концепцией служб транспортного уровня семейства протоколов TCP/IP является концепция *портов*, представляющих собой 16-битный номер и идентифицирующих службу прикладного уровня стека протоколов TCP/IP.

6.2. Протокол UDP

Относить *протокол пользовательских датаграмм (User Datagram Protocol, UDP)* (см. RFC 768[1]) к транспортному уровню не вполне корректно. *UDP* ненадёжен в том смысле, что доставка пакетов не гарантируется. Протокол *UDP* — это протокол *без установления соединения (ConnectionLess)*. Он не устанавливает виртуального соединения, не осуществляет никаких повторных передач, не выполняет переупорядочивания пакетов, не управляет потоком данных. Все эти функции возложены на протоколы более высокого уровня (или приложения).

Формат заголовка пакета UDP показан на рис. 6.1. Поле данных на рисунке не показано. Заметим лишь, что оно выравнивается по 32-битной границе нулевыми байт-заполнителями.

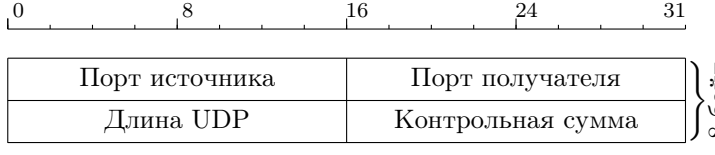


Рис. 6.1. Формат заголовка пакета UDP

Поля *Порт источника (Source Port)* (длина 16 бит) и *Порт получателя (Destination Port)* (длина 16 бит) идентифицируют передающий и получающий процессы соответственно.

[1] *Postel J.* User Datagram Protocol, RFC 768. URL: `http://www.faqs.org/rfcs/rfc768.html`.

Поле *Длина UDP (Length)* (длина 16 бит) содержит длину пакета UDP в байтах.

Поле *Контрольная сумма UDP (Checksum)* (длина 16 бит) содержит контрольную сумму пакета UDP, вычисляемую по всему пакету UDP с добавленным псевдозаголовком (рис. 6.2).

Рис. 6.2. Структура пакета UDP при вычислении контрольной суммы

Во время вычисления контрольной суммы это поле выставляется в нуль, а поле данных выравнивается по 32-байтной границе нулевыми байтами. Если контрольная сумма в полученном пакете равняется нулю, то считается, что передающий уровень UDP её не вычисляет, и данные не защищены.

Псевдозаголовок формируется исключительно для работы с контрольной суммой и имеет следующую структуру (рис. 6.3).

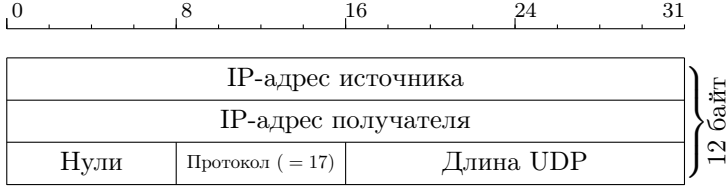


Рис. 6.3. Структура псевдозаголовка пакета UDP

Вначале идут поля *IP-адрес источника* (длина 32 бит) и *IP-адрес получателя* (длина 32 бит).

Далее идёт зарезервированное поле (длина 8 бит), заполненное нулями.

Поле *Протокол* (длина 8 бит) идентифицирует протокол из заголовка пакета IP. Для UDP это значение равно 17 (см. табл. 5.1).

Далее идёт поле *Длина UDP* (длина 16 бит).

Защита заголовка IP несколько избыточна и делает протокол UDP (впрочем, как и TCP) неотделимым от протокола IP, хотя это и поз-

воляет провести двойную проверку датаграмм IP, поступивших для заданного получателя.

Протоколом UDP пользуются приложения, которым нужно передавать датаграммы последовательно. Например, это такие протоколы, как *протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP)*, *служба именования доменов (Domain Name Service, DNS)*, *простой протокол управления сетью (Simple Network Management Protocol, SNMP)* и др. Пользуясь UDP, приложение несёт ответственность за коррекцию ошибок.

6.3. Протокол TCP

Протокол управления передачей (Transmission Control Protocol, TCP) является, в отличие от UDP, «настоящим» протоколом транспортного уровня, который имеет средства управления потоком и коррекции ошибок. Он ориентирован на установление соединения (RFC 793[1])[2].

6.3.1. Формат пакета TCP

На рис. 6.4 показана структура заголовка сегмента TCP.

Рис. 6.4. Формат заголовка пакета TCP

Поля *Порт источника (Source Port)* (длина 16 бит) и *Порт получателя (Destination Port)* (длина 16 бит) аналогичны таким же полям в заголовке пакета UDP (см. раздел 6.2) и идентифицируют процесс или приложение, использующее протокол TCP.

[1] *Postel J.* Transmission Control Protocol, RFC 793. URL: `http://www.faqs.org/rfcs/rfc793.html`.

[2] Также в следующих RFC: 1323, 1644, 2018, 2581, 2582, 2861, 2873, 2883, 2923, 2988, 3293, 3448, 3465, 3481.

Поля *Порядковый номер (Sequence Number)* (длина 32 бита) и *Номер подтверждения (Acknowledgement Number)* (длина 32 бита) нумеруют каждый отправленный или полученный байт данных. Эти поля реализуются как целые числа без знака, которые сбрасываются, когда достигают максимального значения. Каждая сторона ведёт собственную порядковую нумерацию.

Поле *Длина заголовка (Offset)* (длина 4 бита) содержит размер TCP-заголовка в 32-битных словах. Эта информация необходима, так как поле *Параметры (Option)* может быть переменной длины. Можно сказать, что это поле задаёт смещение от начала сегмента до начала данных в 32-битных словах.

Следом идёт неиспользуемое поле (Resrvd) длиной 6 бит. Затем идёт поле *Флаги* длиной 6 бит (рис. 6.5).

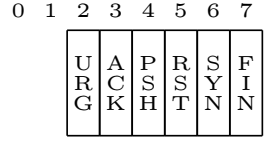


Рис. 6.5. Поле *Флаги* заголовка пакета TCP

Флаг *Указатель срочности (Urgent Pointer, URG)* устанавливается в 1 в случае использования поля *Указатель на срочные данные*.

Флаг *Подтверждение (Acknowledgment, ACK)* устанавливается в 1 в случае, если поле *Номер подтверждения (Acknowledgement Number)* содержит данные. В противном случае это поле игнорируется.

Флаг *Выталкивание (Push, PSH)* означает, что принимающий стек TCP должен немедленно информировать приложение о поступивших данных, а не ждать, пока буфер заполниться. Большинство современных реализаций TCP просто игнорируют флаг *PSH* во время приёма пакетов. Этот флаг оставлен по историческим причинам.

Флаг *Сброс (Reset, RST)* используется для отмены соединения из-за ошибки приложения, отказа от неверного сегмента, попытки создать соединение при отсутствии затребованного сервиса.

Флаг *Синхронизация (Synchronize, SYN)* устанавливается при инициировании соединения и синхронизации порядкового номера.

Флаг *Завершение (Finished, FIN)* используется для разрыва соединения. Он указывает, что отправитель закончил передачу данных.

Управление потоком в протоколе TCP осуществляется при помощи скользящего окна переменного размера. Поле *Размер окна (Window)* (длина 16 бит) содержит количество байт, которое может быть послано после байта, получение которого уже подтверждено. Если значение этого поля равно нулю, это означает, что все байты, вплоть до байта с

номером *Номер подтверждения* – 1, получены, но получатель отказывается принимать дальнейшие данные. Разрешение на дальнейшую передачу выдаётся отправкой сегмента с таким же значением поля *Номер подтверждения* и ненулевым значением поля *Размер окна*.

Поле *Контрольная сумма TCP (Checksum)* (длина 16 бит) содержит контрольную сумму пакета TCP, вычисляемую по всему пакету TCP с добавленным псевдозаголовком (рис. 6.6). Во время вычисления контрольной суммы это поле выставляется в нуль, а поле данных выравнивается по 32-байтной границе нулевыми байтами.

Рис. 6.6. Структура пакета TCP при вычислении контрольной суммы

Псевдозаголовок формируется исключительно для работы с контрольной суммой и имеет следующую структуру (рис. 6.7).

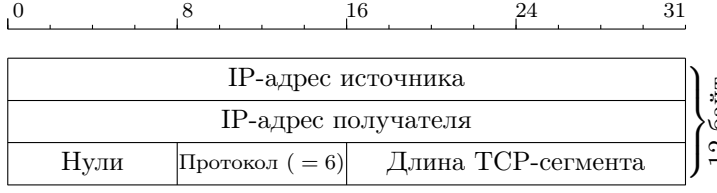


Рис. 6.7. Структура псевдозаголовка пакета TCP

Вначале идут поля *IP-адрес источника* (длина 32 бит) и *IP-адрес получателя* (длина 32 бит).

Далее идёт зарезервированное поле (длина 8 бит), заполненное нулями.

Поле *Протокол* (длина 8 бит) идентифицирует протокол из заголовка пакета IP. Для TCP это значение равно 6 (см. табл. 5.1).

Далее идёт поле *Длина TCP* (длина 16 бит).

Поле *Указатель на срочные данные* (длина 16 бит) содержит смещение в байтах от текущего порядкового номера байта до места расположения срочных данных. Содержимым срочных данных занимаются вышестоящие уровни.

Поле *Параметры (Option)* (длина переменная, кратная 32 битам) содержит дополнительные поля, расширяющие возможности стандартного заголовка. Это поле зарезервировано для будущего применения и в заголовке может отсутствовать. В настоящее время определены опции:

- конец списка опций;
- никаких операций (используется для заполнения поля опции до числа октетов, кратного 4);
- максимальный размер сегмента (Maximum Segment Size, MSS), задающий верхний размер поля данных.

Данные в TCP-сегменте могут и отсутствовать, характер и формат передаваемой информации задаются исключительно прикладной программой, теоретически максимальный размер этого поля составляет в отсутствие опций 65495 байт.

6.3.2. Установление сессии TCP

Поля *Порядковый номер (Sequence Number)* и *Номер подтверждения (Acknowledgment Number)* играют роль счётчика пакетов. При установлении сессии используется поле флагов.

Установление связи клиент-сервер осуществляется в три этапа (трёхступенчатый handshake) (рис. 6.8) .

Пусть хост A создаёт соединение с хостом B.

1. *Режим активного доступа (Active Open)*. Клиент посылает сообщение *SYN, ISSa*, т.е. в передаваемом сообщении установлен бит SYN (Synchronize Sequence Number), а в поле *Порядковый номер (Sequence Number)* — начальное 32-битное значение ISSa (Initial Sequence Number).
2. *Режим пассивного доступа (Passive Open)*. Сервер откликается, посылая сообщение *SYN, ACK, ISSb, ACK(ISSa+1)*, т.е. установлены биты SYN и ACK; в поле *Порядковый номер (Sequence Number)* хостом B устанавливается начальное значение счётчика — ISSb; поле *Номер подтверждения (Acknowledgment Number)* содержит значение ISSa, полученное в первом пакете от хоста A и увеличенное на единицу.
3. *Завершение рукопожатия*. Клиент отправляет подтверждение получения SYN-сегмента от сервера с идентификатором, равным ISN (сервера)+1: *ACK, ISSa+1, ACK(ISSb+1)*. В этом пакете установлен бит ACK, поле *Порядковый номер (Sequence Number)* содержит ISSa + 1, поле *Номер подтверждения (Acknowledgment Number)* содержит значение ISSb + 1. Посылкой этого пакета заканчивается трёхступенчатый handshake, и TCP-соединение считается установленным.

4. Теперь клиент может посылать пакеты с данными на сервер по только что созданному виртуальному TCP-каналу: *ACK, ISSa+1, ACK(ISSb+1); DATA*.

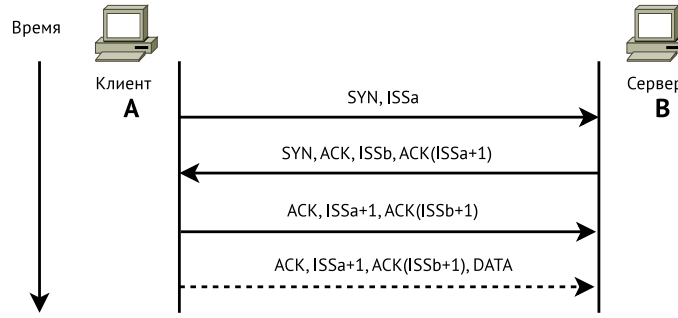


Рис. 6.8. Трёхступенчатый handshake

Из рассмотренной выше схемы создания TCP-соединения видно, что единственными идентификаторами TCP-абонентов и TCP-соединения являются два 32-битных параметра *Порядковый номер (Sequence Number)* и *Номер подтверждения (Acknowledgment Number)*.

6.3.3. Управление потоком

Для ускорения и оптимизации процесса передачи больших объёмов данных протокол TCP определяет метод управления потоком, называемый *методом скользящего окна*, который позволяет отправителю посылать очередной сегмент, не дожидаясь подтверждения о получении в пункте назначения предшествующего сегмента.

Протокол TCP формирует подтверждения не для каждого конкретного успешно полученного пакета, а для всех данных от начала посылки до некоторого порядкового номера *ACK SN (Acknowledge Sequence Number)*. В качестве подтверждения успешного приёма, например, первых n байт, высылается ACK SN $= n + 1$: это означает, что все данные в байтовом потоке под номерами от ISN+1=1 до n успешно получены.

Вместе с посылкой отправителю порядкового номера ACK SN получатель объявляет также размер окна. Это значит, что отправитель может посылать данные с порядковыми номерами от текущего ACK SN до (ACK SN + размер окна - 1), не дожидаясь подтверждения со стороны получателя. Если не будет получено новое подтверждение (новый ACK SN), отправитель будет посылать данные, пока он остаётся в пределах объявленного окна. После этого посылка данных

будет прекращена до получения очередного подтверждения и нового размера окна.

Размер окна выбирается таким образом, чтобы подтверждения успевали приходить вовремя и остановки передачи не происходило. Размер окна может динамически изменяться получателем.

Для временной остановки посылки данных достаточно объявить нулевое окно. Но даже в этом случае через определённые промежутки времени будут отправляться сегменты с одним октетом данных. Это делается для того, чтобы отправитель гарантированно узнал о том, что получатель вновь объявил ненулевое окно, поскольку получатель обязан подтвердить получение пробных сегментов, а в этих подтверждениях он укажет также и текущий размер своего окна. В протоколе TCP скользящее окно используется для регулировки трафика и препятствия переполнению буфера.

Регулирование трафика в TCP подразумевает существование двух независимых процессов: *контроля доставки*, управляемого получателем с помощью параметра *Размер окна (Window)*, и *контроля перегрузки*, управляемого отправителем с помощью *Окна перегрузки (Congestion Window, CWnd)* и *Порога медленного старта (Slow Start Threshold, SSThreth)*.

Первый процесс отслеживает заполнение входного буфера получателя, второй — регистрирует перегрузку канала и связанные с этим потери, а также понижает интенсивность трафика. В исходный момент времени при установлении соединения CWnd делается равным одному MSS (максимальному размеру сегмента), а SSThreth — 65535 байтам. Программа, управляющая пересылкой, никогда не пошлёт больше байт, чем это задано CWnd и объявленным получателем значением *Размера окна (Window)*. Когда получение очередного блока данных подтверждено, значение CWnd увеличивается. Если значение CWnd меньше или равно значению SSThreth, то выполняется процедура *Медленный старт*, в противном случае осуществляется подавление перегрузки. В последнем случае CWnd_i+1 = CWnd_i + MSS/8 +(MSS*MSS)/CWnd. Если возникает состояние перегрузки канала, значение CWnd снова делается равным одному MSS. Окно перегрузки позволяет согласовать полную загрузку виртуального соединения и текущие возможности канала, минимизируя потери пакетов при перегрузке.

Для управления потоком используется *Порог медленного старта (SSThresh)*. При установлении соединения SSThresh=64 Кбайт. В случае возникновения таймаута значение SSThresh становится равным CWnd/2, а само значение CWnd приравнивается MSS. Далее запускается процедура медленного старта, чтобы выяснить возможности канала. При этом экспоненциальный рост CWnd осуществляется вплоть до значения SSThresh. Когда этот уровень CWnd достигнут, дальнейший рост происходит линейно с приращением на каждом шаге, равном MSS.

6.3.4. Проблемы TCP

TCP за годы существования претерпел значительные изменения, касающиеся обеспечения надёжности и производительности в сетях различной ёмкости и качества. Но при этом возможности TCP уже не удовлетворяют современным потребностям.

- TCP не подходит для передачи данных в VoIP-сетях или для асинхронной обработки на базе транзакций.
- TCP требует строго упорядоченной передачи данных, что не подходит для приложений, допускающих как последовательную, так и непоследовательную доставку потоков.
- TCP не структурирует последовательности передаваемых данных. Поэтому требуется добавление разграничителей сообщений.
- TCP не поддерживает множественную адресацию.
- TCP-хосты восприимчивы к атакам «отказ в обслуживании» (Denial of Service, DoS) типа SYN DoS (или FIN DoS)[1].

6.4. Протокол SCTP

Протокол управления потоковой передачей (Stream Control Transmission Protocol, SCTP) (RFC 2960[2], RFC 3286[3]) можно рассматривать как дальнейшее логическое развитие протокола TCP. Как и TCP, протокол SCTP предлагает приложениям, взаимодействующим по IP-сети, ориентированную на соединения типа «точка-точка» транспортную службу с надёжной доставкой. Протокол унаследовал часть функциональности TCP, в том числе возможность контроля перегрузки и восстановления утерянных пакетов. Любое приложение, работающее по протоколу TCP, можно перевести на SCTP без потери функциональности.

[1] Хосту посылается огромное количество пакетов TCP SYN. Хост-получатель резервирует память и отвечает на запрос сообщениями SYN ACK. Когда атакующая система не возвращает сообщения ACK, необходимые для завершения процедуры установки TCP-соединения, ресурсы хоста, подвергнувшегося атаке, остаются неосвобождёнными. Поэтому он оказывается не готов к обслуживанию других запросов.

[2] Stream Control Transmission Protocol, RFC 2960 / R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. URL: `http://www.faqs.org/rfcs/rfc2960.html`.

[3] *Ong L., Yoakum J.* An Introduction to the Stream Control Transmission Protocol (SCTP), RFC 3286. URL: `http://www.faqs.org/rfcs/rfc3286.html`.

6.4.1. Формат пакета SCTP

Сообщения SCTP включают общий заголовок, за которым следует один или несколько *подпакетов (Chunk)*, которые могут содержать данные или управляющую информацию (рис. 6.9). В заголовке (рис. 6.10) указываются номера портов отправителя и получателя, что позволяет мультиплексировать различные ассоциации SCTP на одном адресе.

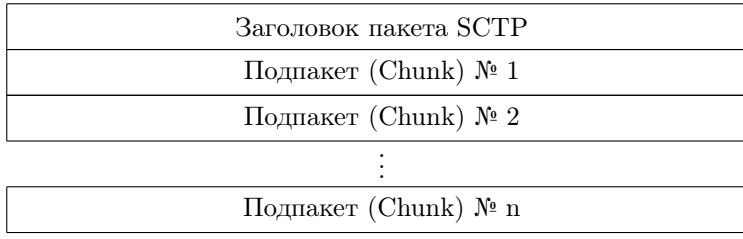


Рис. 6.9. Формат пакета SCTP

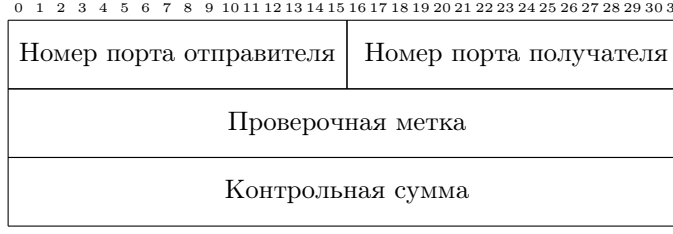


Рис. 6.10. Формат заголовка пакета SCTP

Проверочная метка (Verification Tag) (длина 32 бита) предотвращает возможность включения в ассоциацию SCTP устаревших или фальсифицированных сообщений.

Контрольная сумма (длина 32 бита) рассчитывается на основе полиномиального алгоритма CRC-32c и служит для выявления ошибок.

Формат подпакета

Каждый подпакет (фрагмент) содержит поля (рис. 6.11) *Тип подпакета (Chunk ID)*, *Флаги (Chunk Flags)*, *Длина подпакета (Chunk Length)*, *Данные (Chunk Value)*.

Рис. 6.11. Формат подпакета SCTP

Восьмибитное поле типа подпакета способно принимать до 255 значений (в настоящее время определены 15, а остальные зарезервированы). Если данное поле имеет нулевое значение, то это говорит о передаче *полезной информации (Payload Data)*; в других случаях подпакет несёт служебные сведения.

Второе поле — восьмибитное поле *флагов*, его использование определяется типом подпакета.

Поле *длины* с разрядностью 16 бит заполняется суммарным значением длины подпакета с учётом полей заголовка.

Управляющие блоки включают различные параметры и флаги, зависящие от типа блока. Подпакеты *данных (DATA)* включают флаг управления сегментацией и сборкой, а также параметры *TSN*, *Stream ID*, *Stream Sequence Number* и *Payload Protocol Identifier*.

Перед фрагментом DATA размещаются *номер транспортной последовательности (Transport Sequence Number, TSN)*, *идентификатор потока*, *номер последовательности потока (Stream Sequence Number, SSN)*.

Номер транспортной последовательности используется для обеспечения надёжности каждой ассоциации, а номер последовательности потока — для упорядочивания по потокам. Отдельные сообщения в потоке отмечаются идентификатором потока.

Информационная часть предназначена для передачи собственно данных, которые определяются типом подпакета. Согласно протоколу SCTP, размерность подпакета должна быть кратна 32 битам. В противном случае информационная часть дополняется нулевыми значениями, но в поле длины указывается истинная величина. Это позволяет на приёмной стороне соединения исключить добавленные нули из передаваемых данных.

Параметр *Payload Protocol ID* включён для обеспечения возможности расширения в новых версиях протокола. Если предположить, что функции идентификации протокола и мультиплексирования по портам в будущем перестанут играть столь важную роль, как сейчас, *Payload Protocol ID* будет обеспечивать идентификацию протоколов,

передаваемых с помощью SCTP без использования номера порта.

Формат сообщений SCTP обеспечивает механизм связывания множества блоков данных и управления в одно сообщение для повышения эффективности транспорта. Использованием такой *группировки (Bundling)* управляет приложение, поэтому группировка стартовой передачи невозможна. Связывание естественным образом осуществляется при повторе передачи блоков *DATA* в целях снижения вероятности насыщения.

6.4.2. Функции SCTP

SCTP представляет собой unicast-протокол, который обеспечивает обмен данными между двумя конечными точками.

Аналогом TCP-соединения для SCTP является ассоциация, которая устанавливается между двумя оконечными устройствами. При этом одно устройство может быть определено несколькими IP-адресами, список которых передаётся при установлении ассоциации. Для передачи данных через ассоциацию используются все возможные комбинации адресов пары оконечных устройств.

Отказоустойчивость в таком случае обеспечивается за счёт того, что разные IP-адреса присваиваются различным интерфейсам устройств, и трафик между ними передаётся по разным маршрутам. В случае отказа какого-либо оборудования в сети и недоступности одного или нескольких IP-адресов трафик продолжает передаваться между оставшимися адресами, и разрыва SCTP-ассоциации не происходит.

Описанный выше механизм работы SCTP-ассоциации носит название *многодомности (SCTP Multi-Homing)*.

К другим ключевым функциям протокола SCTP относятся:

- группировка различных сигнальных сообщений в одном пакете с одним SCTP/IP-заголовком (Chunk Bundling), что повышает эффективность использования полосы пропускания;
- последовательная доставка сообщений внутри различных потоков, что позволяет избежать ситуации, встречающейся при использовании протокола TCP, когда в случае потери одного пакета остальные задерживаются в буфере до успешной его перепосылки (Head-of-Line Blocking);
- использование контрольных сумм для обеспечения безошибочной передачи пакетов, а также для защиты от атак.

Протокол SCTP поддерживает ряд функций, унаследованных не только от TCP, но и от других протоколов. При этом в нём реализованы и дополнительные функции:

- *Сохранение границ сообщений.* Сообщения, передаваемые SCTP, размещаются в подпакетах (или фрагментах), что даёт возможность приложениям отделить одно сообщение от другого.

– *Отсутствие блокировок типа head-of-line.* В отличие от TCP протокол SCTP не требует строгой упорядоченности передаваемых пакетов. Поэтому в нём отсутствует задержка, вызываемая блокировкой обслуживания, возникающей при восстановлении TCP корректной последовательности пакетов.
– *Несколько режимов доставки.* SCTP может передавать данные как в строгом порядке (как TCP), так и частично упорядоченные (по потокам) и неупорядоченные вовсе (как UDP).
– *Поддержка многодомности.* SCTP может переадресовывать пакеты на альтернативный IP-адрес.
– *Контроль перегрузки.* SCTP использует стандартные методики, применяющиеся для контроля перегрузки в TCP, в том числе медленный старт, предотвращение перегрузки и быструю повторную передачу.
– *Выборочные подтверждения.* SCTP использует схему выборочного подтверждения, унаследованную из TCP, для восстановления утраченных пакетов.
– *Фрагментация пользовательских данных.* SCTP разбивает сообщения на фрагменты, чтобы *максимальный размер передаваемого элемента (Maximum Transfer Unit, MTU)* соответствовал ограничениям конкретного маршрута пересылки между взаимодействующими хостами (RFC 1191[1]).
– *Механизм контроля работоспособности (Hearbeat).* SCTP посылает пакеты контроля работоспособности на адреса находящегося в режиме ожидания хоста, которые входят в ассоциацию. Протокол декларирует, что IP-адрес будет отключён, как только он достигнет порогового значения невозвращённых подтверждений о работоспособности.
– *Защита от DoS-атак.* SCTP использует механизм cookie при инициализации ассоциации, чтобы смягчить воздействие DoS-атак.

6.4.3. Множественность потоков и варианты доставки

Название протокола SCTP обусловлено его многопотоковой природой передачи данных. Поддержка множества одновременных потоков позволяет распределить между этими потоками передаваемую информацию так, чтобы каждый из потоков обеспечивал независимую упорядоченную доставку данных. Потеря сообщения в любом из потоков оказывает влияние лишь на данный поток, не затрагивая работу других потоков данных.

Протокол TCP работает с одним потоком данных и обеспечивает сохранение порядка доставки байт из потока. Такой подход удобен

[1] *Mogul J., Deering S.* Path MTU discovery, RFC 1191. URL: `http://www.faqs.org/rfcs/rfc1191.html`.

для доставки файлов или записей, но он может приводить к дополнительным задержкам при потере информации в сети или нарушении порядка доставки пакетов. При возникновении подобных ситуаций протокол TCP должен дождаться доставки всех данных, требуемых для восстановления порядка.

В рамках одного соединения SCTP обеспечивает единый механизм управления потоком и контроля насыщения, что существенно снижает нагрузку на транспортный уровень.

SCTP разделяет понятия надёжной и упорядоченной доставки, в то время как в TCP эти два аспекта неразрывно связаны, так как все данные надёжно доставляются хосту-получателю и предоставляются приложению в той последовательности, в какой они передавались. Для этого TCP использует номер последовательности в заголовке каждого пакета.

Протокол SCTP поддерживает многопотоковую передачу за счёт устранения зависимости между передачей и доставкой данных. В частности, каждый блок полезной информации типа DATA (данные) использует два набора порядковых номеров. Номер TSN управляет передачей сообщений и детектированием их потери, а пара *идентификатор потока Stream ID–номер SSN* используется для управления порядком доставки потребителю полученных данных.

Такая независимость механизмов нумерации позволяет получателю незамедлительно обнаруживать пропуски данных, а также видеть влияние потерянных данных на поток. Утрата сообщения вызывает появление пропуска в порядковых номерах SSN для потока, на который это сообщение оказывает влияние и не вызывает такого пропуска для других потоков. Следовательно, получатель может продолжить доставку незатронутых потоков, не дожидаясь повтора передачи утраченного сообщения.

6.4.4. Многодомность

Механизм многодомности предназначен для повышения устойчивости сети к выходам из строя интерфейсов на хосте и ускорения восстановления в случае сбоя в сети. Но эффективность этого механизма падает, если путь взаимодействия внутри ассоциации проходит через единую точку сбоя сети.

Действующий вариант SCTP не поддерживает *распределения нагрузки (Load Sharing)*, поэтому многодомные хосты обеспечивают лишь избыточность соединений для повышения уровня надёжности. Один из адресов многодомного хоста указывается в качестве *основного (Primary)* и используется как адрес получателя для всех блоков данных при нормальной передаче. При передаче повторных блоков данных используется один из дополнительных адресов с целью повышения вероятности доставки в конечную точку. При повторяющихся неоднократно повторах передачи принимается решение об отправке

всех блоков данных с использованием альтернативного адреса, пока системе мониторинга не удастся увидеть доступность основного адреса.

Для поддержки множества интерфейсов конечные точки SCTP обмениваются списками своих адресов в процессе создания ассоциации. Каждая из конечных точек должна быть способна принимать сообщения с любого адреса, связанного с удалённым партнёром; на практике некоторые операционные системы могут использовать в пакетах циклический перебор адресов отправителя, и в таких случаях приём пакетов с различных адресов является нормальной ситуацией. Для всего списка адресов конечной точки в данной сессии используется один номер порта.

Для повышения уровня безопасности требуется, чтобы некоторые отклики передавались по адресу, указанному в поле отправителя сообщения, вызвавшего отклик. Например, когда сервер получает блок INIT от клиента для инициирования SCTP-ассоциации, сервер всегда будет передавать блок INIT ACK по адресу отправителя в заголовке IP-блока INIT.

6.4.5. Установление ассоциаций

SCTP, как и TCP, ориентирован на установление соединения. Оба протокола требуют определения состояния соединения на каждом хосте. Соединение TCP определяется двумя IP-адресами и двумя номерами портов. Ассоциация SCTP определяется как набор IP-адресов + порт на каждом хосте. Любые из IP-адресов на любом хосте могут указываться в качестве отправителя или получателя в IP-пакете и это корректно идентифицирует ассоциацию.

Перед началом обмена данными два SCTP-хоста должны передать друг другу информацию о состоянии соединений с помощью четырёхэтапной процедуры установки соединения (handshake) (рис. 6.12).

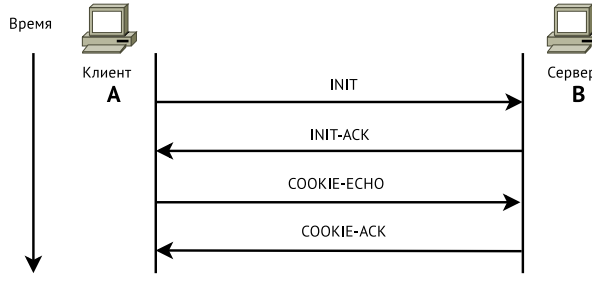


Рис. 6.12. Четырёхэтапная процедура установки соединения SCTP

Процедура, предусмотренная протоколом SCTP, позволяет защититься от DoS-атак. Получателю сообщения о намерении установить контакт *INIT* в четырёхэтапной процедуре установки соединения не требуется сохранять никакую информацию о состоянии или резервировать какие-либо ресурсы. Вместо этого он посылает в ответ сообщение *INIT-ACK*, которое включает в себя *запись состояния (Cookie)*, содержащую всю информацию, необходимую отправителю *INIT-ACK* для того, чтобы сформировать своё состояние. Запись состояния подписывается цифровой подписью.

Оба сообщения, *INIT* и *INIT-ACK*, содержат несколько параметров, необходимых для установки начального состояния:

- список всех IP-адресов, которые станут частью ассоциации;
- номер транспортной последовательности, используемый для надёжной передачи данных;
- тег инициации, который должен быть включён в каждый входящий пакет SCTP;
- число выходящих потоков, запрашиваемых каждой из сторон;
- число входящих потоков, которые способна поддерживать каждая из сторон.

После обмена этими сообщениями, отправитель INIT возвращает назад запись состояния в виде сообщения COOKIE-ECHO, которое также может содержать связанные с ним пользовательские сообщения DATA. При получении COOKIE-ECHO получатель полностью меняет своё состояние и отправляет обратное сообщение COOKIE-ACK, подтверждающее завершение настройки. COOKIE-ACK также может сопровождаться пользовательскими сообщениями DATA.

6.4.6. Завершение работы ассоциации

Транспортному протоколу, ориентированному на соединение, необходим метод постепенного отключения ассоциации. SCTP использует процедуру установки соединения, отличающуюся от процедуры, применяемой в TCP: конечная точка TCP может инициировать процедуру отключения, сохраняя открытым соединение и получая новые данные от другого хоста. SCTP не поддерживает такого наполовину закрытого состояния, т.е. обе стороны не могут передавать новые данные на свой более высокий уровень, если инициирована последовательность постепенного отключения.

Пусть приложение на хосте A хочет отключить и закрыть ассоциацию с хостом B. SCTP устанавливает состояние SHUTDOWN_PENDING, в котором он не будет принимать данные от приложения, но по-прежнему будет посылать новые данные, помещаемые в очередь на передачу на хост B. После подтверждения всех размещённых в очереди данных хост A посылает подпакет SHUTDOWN и устанавливает состояние SHUTDOWN_SENT.

До получения подпакета SHUTDOWN хост B уведомляет свой более высокий уровень, что прекращает принимать от него новые данные и вводит состояние SHUTDOWN_RECEIVED. Хост B передаёт оставшиеся данные на A, за которыми следуют фрагменты SHUTDOWN, информирующие B о появлении данных и подтверждающие, что ассоциация отключена. Как только подтверждены все данные, помещённые в очередь на хосте B, хост A посылает соответствующий фрагмент SHUTDOWN-ACK, за которым следует фрагмент SHUTDOWN-COMPLETE, завершающий отключение ассоциации.

6.5. Протокол DCCP

Протокол DCCP (Datagram Congestion Control Protocol)[1] является транспортным протоколом, который использует двунаправленные уникастные соединения с управлением перегрузкой для ненадёжной доставки дейтаграмм.

Протокол DCCP имеет встроенную систему управления перегрузкой, включающую поддержку *уведомления о перегрузке канала (Explicit Congestion Notification, ECN)*[2] для ненадёжных потоков дейтаграмм, исключая непредсказуемые задержки, характерные для TCP, что обеспечивает надёжное согласование параметров при установлении соединения.

6.5.1. Характеристики DCCP

Протокол DCCP обладает следующими характеристиками:
- является протоколом для потоков пакетов, а не потоков байт;
- реализует поток дейтаграмм с подтверждением получения, но без повторной посылки;
- имеет ненадёжный диалог установления и разрыва соединения;
- обеспечивает надёжное согласование параметров;
- предоставляет выбор механизмов подавления перегрузки;
- является протоколом управления перегрузкой, а не протоколом управления потоками;
- имеет опции, указывающие отправителю, был ли пакет доставлен получателю, помечен ECN, повреждён или отброшен входным буфером получателя;

[1] *Floyd S.*, *Handley M.*, *Kohler E.* Problem Statement for the Datagram Congestion Control Protocol (DCCP), RFC 4336. URL: `http://tools.ietf.org/html/rfc4336`; *Kohler E.*, *Handley M.*, *Floyd S.* Datagram Congestion Control Protocol (DCCP), RFC 4340. URL: `http://tools.ietf.org/html/rfc4340`.

[2] *Ramakrishnan K.*, *Floyd S.*, *Black D.* The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168. URL: `http://tools.ietf.org/html/rfc3168`.

– осуществляет управление перегрузкой со встроенной индикацией явной перегрузки ECN;
– обладает механизмами, позволяющими серверу избежать поддержки состояний неподтверждённых попыток соединений;
– выявляет MTU пути.

6.5.2. Типы сообщений DCCP

Протокол DCCP использует девять различных типов сообщений:
– DCCP-Request инициирует соединение;
– DCCP-Response является ответом на запрос DCCP-Request;
– DCCP-Data передаёт данные;
– DCCP-Ack передаёт подтверждения о получении пакетов;
– DCCP-DataAck передаёт данные в сочетании с подтверждениями;
– DCCP-CloseReq запрашивает закрытие соединения;
– DCCP-Close осуществляет закрытие соединения или запускает процедуру сброса соединения (DCCP-Reset);
– DCCP-Reset осуществляет процедуру сброса соединения;
– DCCP-Sync, DCCP-SyncAck осуществляют повторную синхронизацию номеров пакетов после длительного периода потерь.

6.5.3. Формат заголовка DCCP

Базовый заголовок DCCP имеет следующий формат (рис. 6.13).

Рис. 6.13. Формат базового заголовка DCCP

Поля *Порт отправителя (Source Port)* и *Порт получателя (Dest Port)* (длиной по 16 бит каждый) идентифицируют соединение. Когда соединение формируется, клиент должен выбрать порт отправителя случайным образом, чтобы уменьшить вероятность атаки.

Поле *Смещение данных (Data Offset)* (длина 8 бит) указывает смещение от начала заголовка пакета DCCP первого октета данных (выражается в 32-битных словах).

Поле *CCVal* (длина 4 бита) используется отправителем CCID.

Поле *Checksum Coverage* (CsCov) (длина 4 бита) определяет части пакета, которые покрываются полем *Контрольная сумма.*

Поле *Контрольная сумма (Checksum)* (длина 16 бит) содержит контрольную сумму заголовка пакета DCCP (включая опции), псевдозаголовка сетевого уровня и, в зависимости от CsCov, полей данных приложений.

Поле *Зарезервировано (Reserved)* (длина 3 бита) содержит нули, получатель должен это поле игнорировать.

Поле *Tun (Type)* (4 бита) специфицирует тип пакета.

Поле *Расширенные порядковые номера (X)* (длина 1 бит) равно нулю, если передаются только младшие (LSB) 24 бита порядкового номера, а базовый заголовок имеет длину 12 байт и значение 1, если в заголовке используются 48-разрядные порядковые номера. Пакеты DCCP-Data, DCCP-DataAck и DCCP-Ack могут иметь значение, X равное 0 или 1. Все пакеты DCCP-Request, DCCP-Response, DCCP-CloseReq, DCCP-Close, DCCP-Reset, DCCP-Sync и DCCP-SyncAck должны иметь X=1.

Поле *Порядковый номер (Sequence Number)* (длина 48 или 24 бита) идентифицирует пакет в последовательности. Номер по порядку увеличивается на 1 после посылки каждого пакета, включая пакеты DCCP-Ack, которые не несут в себе данных.

После базового заголовка следует заголовок пересылаемого типа пакета.

6.5.4. Процедура взаимодействия

Процедура взаимодействия двух элементов следующая.

1. Клиент посылает серверу запрос DCCP-Request на установление соединения. Определяются номера портов клиента и сервера, запрашиваемая услуга и другие параметры соединения, включая CCID, необходимый серверу при работе с клиентом.
2. В ответ сервер посылает пакет-отклик.
3. Клиент посылает серверу подтверждение DCCP-Ack получения DCCP-отклика.
4. Далее по необходимости происходит обмен подтверждениями DCCP-Ack для согласования используемых параметров.
5. Сервер и клиент обмениваются пакетами DCCP-Data, DCCP-Ack.
6. Для закрытия соединения сервер посылает DCCP-CloseReq.
7. Для подтверждения закрытия соединения клиент посылает DCCP-Close.
8. Сервер посылает пакет DCCP-Reset, при этом состояние соединения ликвидируется.
9. Клиент получает пакет DCCP-Reset и сохраняет своё состояние в течение некоторого времени для завершения происходящих обменов.

6.5.5. Функциональность DCCP

Протокол DCCP может реализовать механизм контроля за перегрузкой, многодомность и мобильность (за счёт механизма переадресации), процедуру медленного получателя (Slow Receiver). DCCP не предоставляет криптографических гарантий безопасности, но имеет возможности противостоять некоторым видам атак благодаря используемой системе нумерации пакетов.

6.6. Краткие итоги раздела

1. На транспортном уровне организована служба надёжной доставки данных для верхних уровней, использующая управление потоком и коррекцию ошибок в сквозном потоке.
2. Адресом транспортного уровня является номер порта — 16-битный номер, идентифицирующих службу прикладного уровня стека протоколов TCP/IP.
3. На транспортном уровне семейства протоколов TCP/IP применяются два основных протокола — ориентированный на соединение протокол TCP и не требующий соединения протокол UDP.
4. Протокол *UDP* — это протокол *без установления соединения (ConnectionLess)*. Он не устанавливает виртуального соединения, не осуществляет никаких повторных передач, не выполняет переупорядочивания пакетов, не управляет потоком данных. Все эти функции возложены на протоколы более высокого уровня (или приложения).
5. Протокол TCP имеет средства управления потоком и коррекции ошибок. Он ориентирован на установление соединения, поэтому клиент обязан установить соединение с сервером до начала передачи данных TCP в любом из направлений. Управление потоком в протоколе TCP осуществляется при помощи скользящего окна переменного размера. Установление связи клиент-сервер осуществляется с помощью процедуры «трёхступенчатый handshake».
6. Протокол SCTP унаследовал часть функциональности TCP, в том числе возможность контроля перегрузки и восстановления утерянных пакетов. Любое приложение, работающее по протоколу TCP, можно перевести на SCTP без потери функциональности. Аналогом TCP-соединения для SCTP является ассоциация, которая устанавливается между двумя оконечными устройствами. При этом одно устройство может быть определено несколькими IP-адресами, список которых передаётся при установлении ассоциации. Для передачи данных через ассоциацию используются все возможные комбинации адресов пары оконечных устройств. Отказоустойчивость обеспечивается механизмом многодомности — SCTP может переадресовывать пакеты на альтернативный IP-адрес. SCTP не требует строгой упорядоченности передаваемых пакетов.

7. SCTP разделяет понятия надёжной и упорядоченной доставки. SCTP поддерживает многопотоковую передачу за счёт устранения зависимости между передачей и доставкой данных. Установление связи клиент-сервер осуществляется с помощью процедуры «четырёхступенчатый handshake».
8. *Протокол DCCP* является транспортным протоколом, который использует двунаправленные уникастные соединения с управлением перегрузкой для ненадёжной доставки дейтаграмм. DCCP имеет встроенную систему управления перегрузкой, включающую поддержку уведомления о перегрузке канала (ECN), что обеспечивает надёжное согласование параметров при установлении соединения. DCCP может реализовать механизм контроля за перегрузкой, многодомность и мобильность (за счёт механизма переадресации), процедуру медленного получателя (Slow Receiver). DCCP не предоставляет криптографических гарантий безопасности, но имеет возможности противостоять некоторым видам атак благодаря используемой системе нумерации пакетов.

6.6.1. Вопросы к главе 6

1. Укажите функции и услуги транспортного уровня модели ISO/OSI.
2. В чем состоят принципиальные отличия протоколов TCP и UDP.
3. Опишите схему управления потоком в протоколе TCP.
4. Опишите схему установления сессии TCP.
5. В чем заключаются основные отличия протоколов TCP и SCTP?
6. Опишите функциональность протокола SCTP.
7. В чем заключается механизм многодомности в протоколе SCTP?
8. Опишите схему установления сессии SCTP.
9. Укажите основные характеристики протокола DCCP.
10. В чем заключаются основные отличия протокола DCCP от других транспортных протоколов?
11. Опишите схему установления сессии DCCP.

6.7. Примеры заданий

Задание 6.1 — Для чего, кроме как подсчёта пакетов, могут служить поля *Порядковый номер (Sequence Number)* и *Номер подтверждения (Acknowledgment Number)* пакета TCP?

Ответ (Задание 6.1) — Эти поля могут идентифицировать сессию TCP и служить меткой безопасности сессии (своего рода паролем).

6.8. Задания для самостоятельной работы

Задание 6.2 — Исследуйте зависимость производительности протокола TCP от размеров окна.

Задание 6.3 — Исходя из свойств, обрисуйте область применения протокола UDP.

Задание 6.4 — Исходя из свойств, обрисуйте область применения протокола TCP.

Задание 6.5 — Исходя из свойств, обрисуйте область применения протокола SCTP.

Задание 6.6 — Исходя из свойств, обрисуйте область применения протокола DCCP.

Глава 7. Обеспечение информационной безопасности сетей

Глава посвящена некоторым аспектам информационной безопасности компьютерных сетей. Приведены примеры конфигурирования коммутаторов, принципы и основы конфигурирования виртуальных локальных сетей.

В основу главы легли материалы из источника [3].

В результате освоения данной темы студент должен:

знать:

- принципы построения сетей передачи данных и настройки сетевого оборудования;
- принципы защиты сетевых элементов (коммутаторов и маршрутизаторов) от несанкционированного доступа;
- принципы организации виртуальных локальных сетей (VLAN);

уметь:

- читать и понимать конфигурационный файл сетевого и коммутационного оборудования;
- применять на практике принципы защиты сетевых элементов (коммутаторов и маршрутизаторов) от несанкционированного доступа;

владеть:

- способностью настраивать сетевое и коммутационное оборудование с учётом аспектов информационной безопасности.

7.1. Общие сведения об информационной безопасности

В сети Интернет, в корпоративных и локальных сетях, по которым передаются пакеты цифровых данных, аудио- и видеоинформации, важно обеспечить **информационную безопасность**. Поскольку сеть Интернет является общедоступной, то ей могут воспользоваться злоумышленники для проникновения во внутренние сети предприятий, на серверы и на конечные узлы пользователей. Злоумышленников, получающих несанкционированный доступ к информации, передаваемой по внутренней сети организации, часто называются хакерами.

Получив доступ во внутреннюю сеть, хакер может реализовать ряд угроз:

- хищение личных данных авторизованных пользователей;
- хищение информации;
- уничтожение или изменение данных;
- нарушение нормальной работы сети.

Угроза вторжения во внутреннюю сеть исходит от злоумышленников, расположенных как внутри, так и за пределами организации, использующей сеть передачи данных. При реализации внешних угроз хакеры совершают атаки обычно из Интернета, по беспроводным сетям. Внутренние угрозы исходят от пользователей, которые имеют санкционированный доступ в сеть. В ряде случаев легальные пользователи сетевых услуг вносят угрозу во внутреннюю сеть организации несознательно, например, при копировании зараженных вирусом файлов.

Для предотвращения **несанкционированного использования информации**, ее **изменения** и **отказа доступа** к ней принимается комплекс мер информационной безопасности. Основные положения информационной безопасности регламентируются стандартом ISO/IEC 27002, который был издан Международной организацией стандартизации — МОС (International Organization for Standardization — ISO) и Международной электротехнической комиссией — МЭК (International Electrotechnical Commission — IEC). До 2007 г. это был стандарт ISO/IEC 17799.

Стандарт ISO/IEC 27002 определяет **информационную безопасность** как «сохранение **конфиденциальности** (уверенности в том, что информация доступна только тем, кто уполномочен иметь такой доступ), **целостности** (гарантии точности и полноты информации, а также методов её обработки) и **доступности** (гарантии того, что уполномоченные пользователи имеют доступ к информации и связанным с ней ресурсам)».

Система мер информационной безопасности включает антивирусное программное обеспечение, управление доступом к сети и файлам (пароли, средства аутентификации, позволяющие установить подлинность личности), шифрование передаваемой по сети информации, системы обнаружения и предотвращения вторжения, средства физической безопасности. Следует отметить, что ни один из методов, устройств или средств не способен реализовать надежную защиту информации. Только комплекс мер может обеспечить безопасность требуемого уровня. Ряд принципов или комплекс мер по защите информации называется **политикой безопасности**. Рекомендации по разработке политик и процедур безопасности представлено в Руководстве RFC 2196 (Безопасность сайта, справочник).

В настоящем курсе рассматриваются специфические средства защиты сетей и систем передачи информации.

Передаваемая по сети информация подвергается атакам, среди которых можно выделить **атаки доступа, модификации, отказа в обслуживании**.

Атака доступа производится для получения неавторизованным пользователем (злоумышленником, хакером) не предназначенной ему конфиденциальной информации. **Пассивная атака** доступа реализуется путем подсматривания (**snooping**) или подслушивания (**sniffing**)

интересующей злоумышленника информации, проходящей по сети. **Прослушивание** легко реализуется в сетях с разделяемой средой передачи, т.е. в сетях с концентраторами (хабами), а также в беспроводных сетях.

Например, злоумышленник может подключиться к сети через концентратор (рис. 7.1) и прослушивать проходящий по концентратору трафик. Для этого взломщик устанавливает в компьютерной системе сетевой анализатор пакетов («снифер», **sniffer**). Анализ трафика в ходе такой пассивной атаки проводится, прежде всего, для определения имен и паролей авторизованных пользователей, зная которые можно полностью контролировать передаваемую информацию.

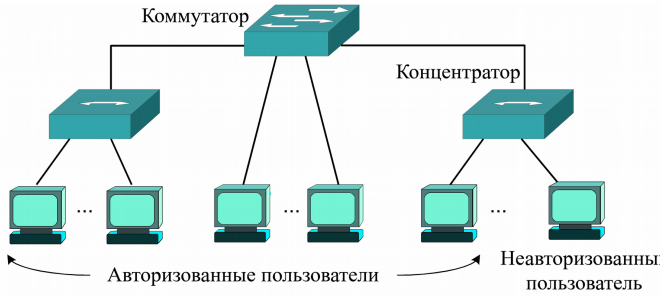


Рис. 7.1. Подключение к сети неавторизованного пользователя

В настоящее время в локальных сетях широко используются коммутаторы, когда принятые кадры не рассылаются широковещательно во все порты, а продвигаются непосредственно на порт узла-получателя. Однако хакер может обойти это препятствие. Наводнив коммутатор ложными сообщениями с MAC-адресами несуществующих источников, хакер переполняет таблицу коммутации и вынуждает коммутатор работать в режиме концентратора. При этом злоумышленнику будет доступен весь проходящий по коммутатору трафик. Причем, авторизованный пользователь может этого даже не заметить.

При активной атаке производится **перехват трафика** и после анализа перехваченной информации хакер решает вопрос о ее дальнейшей передаче адресату назначения или уничтожении. Для этого хакер может перенаправить трафик коммутатора к «сниферу». Перенаправление трафика путем подмены адреса назначения передаваемого кадра получило название «спуфинг» (**spoofing**). Например, при формировании кадра производится ARP-запрос для получения MAC-адреса назначения по заданному IP-адресу. Однако такой запрос может захватить установленный в сети «снифер», который на запрос ответит собственным MAC-адресом, поэтому трафик будет

перенаправлен злоумышленнику.

Хакер может также продублировать MAC-адрес атакуемого узла на «снифере», поэтому весь трафик, предназначенный легальному узлу, будет дублироваться «сниферу».

Перехваченная информация затем может быть уничтожена, искажена или без искажения передана адресату назначения. **Атака модификации** – это **неправомочное изменение информации,** т.е. **нарушение целостности информации**. При этом производится либо **замена** передаваемой информации, либо **добавление** новых данных, либо **удаление** старых передаваемых данных. Для реализации такой атаки необходимо предварительно выполнить перехват передаваемой информации, затем провести ее модификацию и передать на узел назначения.

Например, «снифером» может быть реализован перехват DNS-запросов, когда он на запрос отвечает IP-адресом. Поэтому «снифер» получает возможность перехвата всей предназначенной атакуемому узлу информации. Перехваченная информация анализируется, при необходимости модифицируется, и затем отправляется легальному получателю.

При такой атаке для «снифера» важно, чтобы он получил DNS-запрос и ответил на него раньше легального устройства. Поэтому для «снифера» желательно, чтобы он располагался в той же локальной сети, что и узел отправитель. При атаке через Интернет заранее неизвестно, кто быстрее ответит на запрос — «снифер» или легальный DNS-сервер. Перенаправление трафика путем подмены или дублирования адреса назначения легко реализуется, если «снифер» злоумышленника находится в той же локальной сети, что и атакуемый узел. Поэтому очень важна роль физической защиты сети, чтобы предотвратить возможность подключения аппаратуры хакера к локальной сети.

7.1.1. Отказ в обслуживании, DoS-атаки

Атака на **отказ в обслуживании** (**Denial-of-service** — **DoS**) не дает возможность авторизованному легальному пользователю возможность передавать по сети свою информацию. Для этого хакер наводняет (flooding) системы и сети посторонним трафиком, что блокирует доставку легитимного трафика. При реализации такой атаки взломщик организует лавинообразную рассылку данных по сети, например, широковещательных сообщений (запросов). При этом буферы сетевых устройств и конечных узлов переполняются, и вся полоса пропускания линии связи расходуется на пересылку ложных сообщений — сеть «падает». Обычно DoS-атаки запускаются с подложных адресов, поскольку IP-протокол не проверяет адрес источника информации, который использовался при создании пакета.

Широко известной DoS-атакой является атака с использованием

сигналов установления соединения протоколом TCP. Установление соединения происходит за три этапа. На первом этапе узел отправитель инициализирует установление связи путем посылки узлу получателю запроса синхронизации SYN (1). На втором этапе узел получатель подтверждает запрос синхронизации и задает свои параметры синхронизации ACK (2). При этом информацию о новом соединении узел получатель помещает в буфер очереди соединений. На третьем этапе узлу получателю посылается подтверждение ACK (3), что обе стороны готовы для передачи данных.

При DoS-атаке хакер в роли узла отправителя не посылает подтверждение ACK (3), а продолжает отправлять новые ложные запросы синхронизации SYN (1). При этом буфер очереди соединений узла получателя переполняется, и узел перестает отвечать на новые запросы. Если в качестве узла получателя выступает сервер, то он прекращает выполнять предписанные ему услуги. Для сокрытия адреса отправителя используются частные адреса из диапазонов 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, когда без знания информации о параметрах транслятора NAT невозможно идентифицировать источник атаки. Еще сложнее определить злоумышленника, если атака организована из нескольких источников.

Другой известной DoS-атакой является атака с использованием утилиты Ping. Стандартный пакет Ping модифицируют, добавив в него большой объем данных, который переполняет буфер атакуемого узла и выводит его из строя. Для борьбы с этой атакой достаточно провести проверку размера пакета Ping.

7.1.2. Распределённая DoS-атака (DDoS)

Распределённая DoS-атака (Distributed DoS attacks — **DDoS**) представляет собой более разрушительную модификацию DoS-атаки. Целью подобной атаки является перенасыщение и переполнение сетевых каналов и узлов бесполезными данными. В распределенных DoS-атаках **невольно принимают участие** много узлов. Хакер посылает, например, широковещательные пакеты Ping в какую-то большую сеть. При этом производится подмена адреса-источника (**spoofing**), поэтому все ответы на запрос Ping адресуются узлу, чей адрес отправителя указан в запросах широковещательных пакетов. В результате буфер атакуемого узла переполняется, работа узла нарушается.

Конфиденциальность и целостность передаваемой по сети информации обеспечиваются ее **шифрованием**, чтобы даже перехваченная информация не могла быть доступна не авторизованным пользователям. Устройства шифрования должны быть обеспечены соответствующими **ключами** шифрования.

Конфиденциальность и целостность передаваемой информации могут быть нарушены при различных видах атак:

– атаки методом грубой силы;
– использование шпионского программного обеспечение (ПО).

В атаках методом грубой силы производится подбор паролей и дешифрование зашифрованной информации. Для реализации подобной атаки требуется быстродействующий компьютер, чтобы перебрать большое число вариантов паролей и ключей шифрования. Шпионское ПО собирает персональные данные пользователей (имена, пароли и др.) с конечных узлов. Эти данные могут быть затем использованы в атаках методом грубой силы.

Перехват информации, приводящий к потере ее конфиденциальности и целостности, помогает предотвратить система **идентификации и аутентификации** пользователей. Важным элементом системы идентификации и аутентификации пользователей являются **пароли** и **цифровая подпись**.

Общие правила развертывания и функционирования системы безопасности задаются **политикой**, которая определяет **цель** системы безопасности, **область** ее применения и **ответственность** пользователей.

Для обеспечения сетевой безопасности используют совокупность методов и средств, которые отражают политику сетевой безопасности. Особенно строгие требования к технической безопасности необходимо предъявлять в случае использования беспроводных сетей, поскольку в этом случае наиболее легко получить доступ к атакуемой сети.

Таким образом, для обеспечения информационной безопасности сетей и систем передачи информации необходимо сочетание методов и средств **физической и технической** безопасности. В настоящем курсе рассматриваются некоторые методы и средства обеспечения технической безопасности, среди которых наиболее известными являются: формирование комплекса **паролей**, **межсетевые экраны** (сетевые фильтры, списки доступа), **виртуальные локальные сети**.

7.1.3. Формирование паролей на сетевых элементах

Формирование набора паролей необходимо для защиты сетевых элементов (коммутаторов и маршрутизаторов) от несанкционированного доступа не авторизованных пользователей, что может привести к нарушению конфиденциальности, целостности и доступности информации.

Информация для конфигурирования сетевых элементов (маршрутизаторов и коммутаторов) может приходить от различных источников через разные линии, например:

– с линии консольного порта (Console);
– с виртуальных линий интерфейсов терминалов (Virtual Terminals — **vty 0-4**) при использовании протоколов удаленного доступа **Telnet**

или **SSH**. Цифры 0-4 означают, что можно использовать 5 сессий **Telnet** для удаленного доступа.

Вход с линий происходит в пользовательском режиме, поэтому необходимо, прежде всего, защитить доступ к сетевым элементам с консоли: ограничить физический доступ и разрешить доступ только по паролю.

Второй путь к конфигурационному файлу сетевых элементов — это **удаленный доступ по виртуальным линиям** (**vty**), который также позволяют изменять конфигурацию сетевых элементов и обеспечивать контроль проходящего по ним трафика.

Просмотр текущей конфигурации сетевого элемента возможен по команде **show running-config** (сокращённо **sh run**) из привилегированного режима, переход в который из пользовательского режима также должен быть защищен паролем.

Для обеспечения безопасности линий, через которые осуществляется вход в маршрутизатор или коммутатор, необходимо сконфигурировать пароль пользовательского режима. Например, установка пароля на линию 0 консольного порта (**console 0**) реализуется следующей последовательностью команд:

```
Router_A(config)# line console 0
Router_A(config-line)# password cisco1
Router_A(config-line)# login
```

Защита паролем виртуальных линий **vty 0 4** для организации удаленного доступа Telnet в маршрутизатор осуществляется последовательностью:

```
Router_A(config-line)# line vty 0 4
Router_A(config-line)# password cisco2
Router_A(config-line)# login
```

Следует обратить внимание, что после ввода команды **line** маршрутизатор переходит в режим детального (специального) конфигурирования, когда приглашение изменяет вид `Router_A(config-line)#`. Следует также отметить, что каждая удаленная сессия (каждая линия **vty**) может быть защищена своим отдельным паролем.

После установки паролей система будет запрашивать их у пользователя. Например, когда будет производиться подключение к сетевому элементу `Router_A` через консольный порт, потребуется пароль **cisco1**. При реализации удаленного доступа, например, с маршрутизатора `Router_B` к маршрутизатору `Router_A` с адресом одного из его интерфейсов 192.168.10.1 по команде:

```
Router_B>telnet 192.168.10.1
```

потребуется пароль **cisco2**.

Следует отметить, что **если не установлен пароль на виртуальные линии vty, то реализация удаленного доступа не возможна**.

Для обеспечения авторизованного входа в привилегированный режим используются два пароля: **enable secret** и **enable password**. На маршрутизаторе и (или) коммутаторе устанавливается один (или оба) из этих паролей. После установки пароля система запрашивает его у пользователя, когда вводится команда **enable**. Формат команд установки паролей **cisco3** и **cisco4** для входа в привилегированный режим приведен ниже:

```
Router_A(config)# enable secret cisco3
Router_A(config)# enable password cisco4
```

Пароль **enable secret** по умолчанию **криптографируется**, поэтому является более строгим. Если установлены оба пароля **enable secret** и **enable password**, как в приведенном примере, то система будет реагировать на пароль **cisco3**. Пароль **enable password** по умолчанию не криптографируется, поэтому его, как и другие пароли, можно посмотреть (и подсмотреть), например, по команде просмотра текущей конфигурации **show running-configuration** (сокращенно **sh run**), которая выполняется из привилегированного режима. Ниже приведена часть распечатки этой команды верификации после установки паролей:

```
Router_A# sh run
...
enable secret 5 $1$mERr$hx5rVt7rPNos4wqbXKX7m0
enable password cisco4
!
line con 0
  password cisco 1
  login
line vty 0 4
  password cisco 2
  login
!
end
```

Из распечатки следует, что только пароль (**cisco**), созданный командой **enable secret**, по умолчанию **криптографируется**, остальные пароли представлены в открытой форме.

В ряде случаев для повышения безопасности бывает необходимо режим криптографирования паролей распространить на все виды паролей. Это делается по команде **service password-encryption** в режиме глобального конфигурирования:

```
Router_A(config)# service password-encryption
```

При этом в текущей конфигурации будут следующие изменения:

```
Router_A# sh run
...
service password-encryption
!
```

```
hostname Router_A
!
enable secret 5 $1$mERr$hx5rVt7rPNos4wqbXKX7m0
enable password 7 0822455D0A1654
!
line con 0
  password 7 0822455D0A164545
  login
line vty 0 4
  password 7 0822455D0A164544
  login
!
```

Из распечатки следует, что после ввода команды **service password-encryption** все пароли криптографированы, причем пароль enable secret имеет сложную криптограмму, а остальные пароли — сравнительно простую.

Отмена режима криптографирования паролей производится по команде **no service password-encryption** в режиме глобального конфигурирования:

```
Router_A(config)# no service password-encryption
```

При этом уже сформированные пароли останутся в прежнем криптографированном виде, а вновь вводимые — криптографироваться не будут.

7.2. Межсетевые экраны

7.2.1. Общие сведения о межсетевых экранах

Межсетевые экраны (брандмауэры, **firewall**) производят фильтрацию трафика, разрешая или блокируя его передачу. Принятие решения о разрешении или запрете прохождения трафика через межсетевые экраны производится на основе анализа ряда параметров передаваемых пакетов:

- IP-адреса источника информации;
- IP-адреса назначения;
- MAC-адресов;
- используемых для передачи протоколов;
- номеров портов приложений.

Фильтрация трафика не только улучшает информационную безопасность, но и повышает производительность сети, поскольку отбрасывает нежелательные пакеты. Повышение производительности особенно будет эффективно в случае, если межсетевой экран расположен близко к источнику нежелательного трафика.

Наиболее часто межсетевые экраны размещают на границе между внутренней сетью организации (интранетом) и Интернетом, что

позволяет управлять всем исходящим и входящим трафиком сети (рис. 7.2).

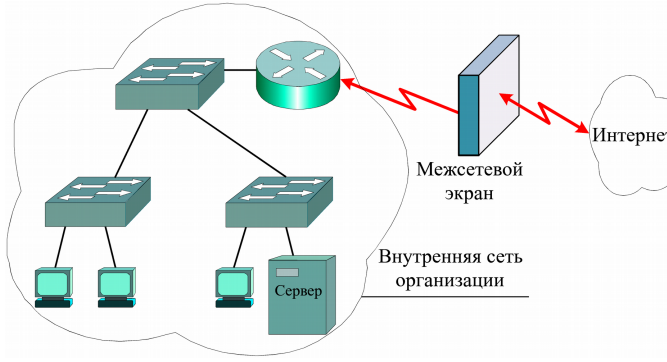


Рис. 7.2. Межсетевой экран между внутренней сетью и Интернетом

Однако в ряде случаев должен быть предусмотрен доступ некоторым внешним пользователям к внутренним ресурсам, например, к некоторым серверам. Для этого обычно формируют демилитаризованную зону (**DMZ**), в которой и располагаются ресурсы, доступные как из внутренней сети организации, так и из Интернета, например Сервер 2 (рис. 7.3).

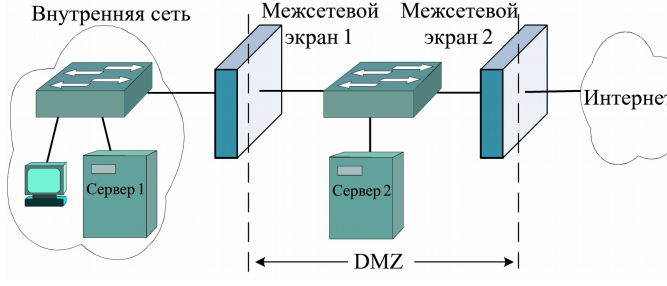


Рис. 7.3. Формирование демилитаризованной зоны

Участок сети DMZ лучше защищен по сравнению с внешней сетью, но его информационная безопасность хуже внутренней сети. В примере рис. 7.3 DMZ сформирована двумя межсетевыми экранами,

которые разграничивают внутреннюю сеть, участок DMZ и внешнюю сеть Интернет. В демилитаризованной зоне DMZ часто размещаются Web-серверы, доступные из внешней сети. В некоторых случаях DMZ формируют на одном межсетевом экране (рис. 7.4).

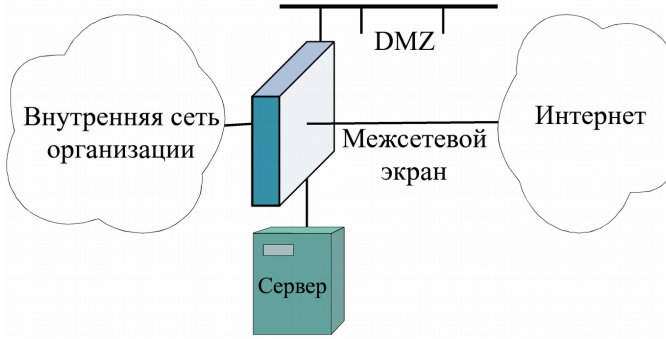


Рис. 7.4. Демилитаризованная зона на одном межсетевом экране

Межсетевые экраны могут быть реализованы в виде отдельных специальных устройств. Кроме того, их функции могут быть встроены в программное обеспечение серверов, а также встроены (интегрированы) в маршрутизаторы. Межсетевые экраны, интегрированные в маршрутизаторы, часто называют **сетевые фильтры** или **списки доступа**. Кроме функций фильтрации трафика, они часто реализуют трансляцию сетевых адресов (NAT). Списки доступа в комплексе с паролями, криптографированием передаваемой информации и физическими устройствами безопасности обеспечивают эффективную защиту сетей нового поколения.

7.3. Списки доступа

Сетевой администратор должен иметь возможность управления потоком данных, обеспечивая доступ к ресурсам сети зарегистрированным пользователям и запрещая нежелательный доступ к сети. Возможности гибкой фильтрации трафика обеспечивают широко применяемые сетевые фильтры или **списки доступа** (Access Lists — **ACL**). Списки доступа могут использоваться, чтобы разрешать (**permit**) или запрещать (**deny**) продвижение пакетов через маршрутизатор, т.е. разрешить или запретить доступ в сеть информации из Интернета, а также удаленный доступ в устройства защищаемой сети.

Списки доступа ACL могут быть созданы для всех сетевых протоколов, функционирующих на маршрутизаторе, например, IP или

IPX, они устанавливаются на интерфейсах маршрутизаторов. Запрет или разрешение сетевого трафика через интерфейс маршрутизатора реализуется на основании анализа совпадения определенных условий. Для этого списки доступа представляются в виде последовательных записей, в которых используют адреса и протоколы. Сетевые фильтры (списки доступа) создаются для входящих или исходящих пакетов на основании анализируемых параметров (адреса источника, адреса назначения, протокола и номера порта), указанных в списке доступа ACL (рис. 7.5).

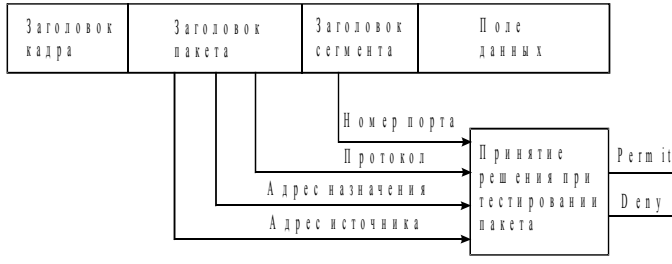


Рис. 7.5. Принятие решения при тестировании пакета

Списки доступа могут быть определены для каждого установленного на интерфейсе протокола и для каждого направления сетевого трафика (исходящего и входящего). Поэтому для входящего и исходящего трафиков через интерфейс создаются отдельные списки. Например, для двух интерфейсов маршрутизатора, сконфигурированных для трех протоколов (IP, AppleTalk и IPX), может быть создано 12 отдельных списков доступа (на каждом интерфейсе по 6 списков, 3 для входящего и 3 для исходящего трафика). Таким образом, для каждого протокола, для каждого направления трафика и для каждого интерфейса может быть создан свой список доступа.

Списки доступа повышают гибкость сети. Например, списки, ограничивающие видео трафик, могут уменьшить нагрузку сети и повысить ее пропускную способность для передачи данных или аудио сигналов. Можно определить, какие типы трафика могут быть отправлены, а какие заблокированы в интерфейсах маршрутизатора, например, можно разрешить маршрутизацию электронной почте, но блокировать трафик Telnet. Можно использовать разрешение или запрет доступа различным типам файлов, таким как FTP или HTTP.

Если списки доступа не формируются на маршрутизаторе, то все проходящие через маршрутизатор пакеты, будут иметь доступ к сети.

Список доступа ACL составляется из **утверждений** (условий),

которые определяют, следует ли пакеты принимать или отклонять во входных и выходных интерфейсах маршрутизатора. Программное обеспечение проверяет пакет последовательно по каждому условию. Если условие, разрешающее продвижение пакета, расположено наверху списка, то **никакие условия, добавленные ниже его, не будут запрещать продвижение пакета**. Если в списке доступа необходимы дополнительные условия, то список целиком должен быть удален и создан новый с новыми условиями.

Функционирование маршрутизатора по проверке соответствия принятого пакета требованиям списка доступа производится следующим образом. Когда кадр поступает на интерфейс, маршрутизатор проверяет MAC-адрес. Если адрес назначения соответствует адресу интерфейса, то маршрутизатор извлекает (декапсулирует) из кадра пакет и проверяет его на соответствие условиям списка ACL. При отсутствии запрета или отсутствии списка доступа пакет инкапсулируется в новый кадр второго уровня и отправляется интерфейсу следующего устройства.

Проверка условий (утверждений) списка доступа производится последовательно. Если текущее утверждение верно, пакет обрабатывается в соответствие с командами **permit** или **deny** списка доступа, остальная часть условий ACL не проверяется. Если все утверждения ACL неверны, то неявно заданная по умолчанию команда **deny any** (запретить все остальное) в конце списка не позволит передавать дальше по сети несоответствующие пакеты.

Существуют разные типы списков доступа: стандартные (standard ACLs), расширенные (extended ACLs) и именованные (named ACLs). Каждый стандартный и расширенный список доступа должен иметь уникальный **идентификационный номер**. Это число идентифицирует тип созданного списка доступа и должно находиться в пределах определенного диапазона, заданного для этого типа списка (табл. 7.1). Именованные списки доступа должны иметь уникальные имена.

Стандартные списки доступа (**Standard access lists**) для принятия решения фильтрования трафика анализируют в IP пакете только адрес источника сообщения.

Расширенные списки доступа (Extended access lists) проверяют как IP-адрес источника, так и IP-адрес назначения, кроме того, анализируют поле протокола в заголовке пакета Сетевого уровня и номер порта в заголовке Транспортного уровня.

Из рекомендаций по установке списков доступа можно отметить следующее. Стандартные списки доступа рекомендуется устанавливать по возможности ближе к адресату назначения, а расширенные — ближе к источнику сообщений. Таким образом, стандартные списки доступа должны защищать устройство назначения и располагаться поближе к защищаемой сети, а расширенные списки доступа должны быть установлены близко к источнику сообщений и не допускать передачу

Таблица 7.1
Диапазоны идентификационных номеров списков доступа

Диапазон номеров	Название списка доступа
1-99	IP standard access-list
100-199	IP extended access-list
1300-1999	IP standard access-list (extended range)
2000-2699	IP extended access-list (extended range)
600-699	Appletalk access-list
800-899	IPX standard access-list
900-999	IPX extended access-list

по сети нежелательного трафика.

Список доступа производит фильтрацию пакетов по порядку записей, поэтому в строках списков следует задавать условия фильтрации, начиная от специфических условий до общих. Условия списка доступа обрабатываются последовательно от вершины списка к основанию, пока не будет найдено соответствующее условие. Если никакое условие не найдено, то тогда пакет отклоняется и уничтожается, поскольку **неявное условие deny any** (запретить все остальное) есть неявно в конце любого списка доступа. Не удовлетворяющий списку доступа пакет протокола IP будет отклонен и уничтожен, при этом отправителю будет послано сообщение ICMP. Новые записи (линии) всегда добавляются в конце списка доступа.

7.3.1. Конфигурирование списков доступа

Конфигурирование списков доступа производится в два этапа:

1. **Создание списка доступа** в режиме глобального конфигурирования.
2. **Привязка списка доступа к интерфейсу** в режиме детального конфигурирования интерфейса.

Формат команды создания стандартного списка доступа следующий:

```
Router(config)# access-list {номер}
  {permit или deny} {адрес источника}.
```

Списки доступа могут фильтровать как трафик, входящий в маршрутизатор (**in**), так и трафик, исходящий из маршрутизатора (**out**). Направление трафика указывается при привязке списка доступа к интерфейсу. Формат команды привязки списка к интерфейсу следующий:

```
Router(config-if)# {протокол} access-group
  {номер} {in или out}
```

После привязки списка доступа его содержимое не может быть изменено. Не удовлетворяющий администратора список доступа должен быть удален командой **no access-list** {номер} и затем создан заново.

Ниже приведены примеры конфигурирования стандартных списков доступа по защите Сети 1 (рис. 7.6).

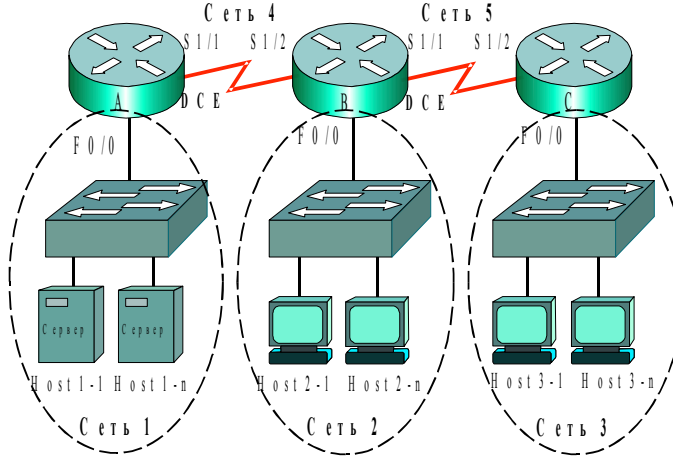


Рис. 7.6. Схема сети

Пример 2. Необходимо, чтобы серверы Сети 1 были доступны только узлу Host 2-1 Сети 2 с адресом 192.168.20.11, а все остальные узлы Сети 2 и Сети 3 не имели бы доступа в Сеть1. Список доступа следует установить на интерфейс F0/0 маршрутизатора `Router_A`. Номер списка доступа (10) выбирается из диапазона табл. 7.1. Адреса сетей, а также названия и адреса интерфейсов приведены в табл. 7.2.

Создание и установка списка доступа производится по командам:

```
Router_A(config)# access-list 10 permit 192.168.20.11
Router_A(config)# int f0/0
Router_A(config-if)# ip access-group 10 out
```

Согласно созданной конфигурации ко всем исходящим из маршрутизатора пакетам через интерфейс F0/0 будет применяться список доступа:

– **permit 192.168.20.11** — присутствует в списке в явном виде,
– **deny any** — присутствует неявно в конце каждого списка доступа.

Таблица 7.2

Адреса сетей и интерфейсов маршрутизаторов

	IP-адрес сети	Интерфейсы	IP-адрес интерфейса
Сеть 1	192.168.10.0/24	F0/0	192.168.10.1
Сеть 2	192.168.20.0/24	F0/0	192.168.20.1
Сеть 3	192.168.30.0/24	F0/0	192.168.30.1
Сеть 4	200.40.40.0/24	S1/1	200.40.40.11
		S1/2	200.40.40.12
Сеть 5	200.50.50.0/24	S1/1	200.50.50.11
		S1/2	200.50.50.12

Некоторые версии операционных систем IOS маршрутизаторов требуют в обязательном порядке использование масок WildCard при задании адресов узлов и сетей, либо расширения **host** при задании адресов узлов. Подобные дополнения рассмотрены в следующих примерах.

Пример 3. Серверы Сети 1 должны быть доступны всем узлам Сети 2 и узлу Host 3-1 Сети 3 с адресом 192.168.30.11, остальные узлы Сети 3 не должны иметь доступа. Список доступа установить на интерфейс F0/0 `Router_A`. В списке доступа имеются адреса сети и отдельного узла, поэтому необходимо использовать маску WildCard. Нулевые значения маски WildCard означают требование обработки соответствующих разрядов адреса, а единичные значения — игнорирование соответствующих разрядов адреса при функционировании списка доступа. Таким образом, маска **0.0.0.0** предписывает анализ и обработку всех разрядов адреса, т.е. в этом случае будет обрабатываться адрес каждого узла. Маска **0.0.0.255** показывает, что обрабатываться будет только сетевая часть адреса класса C.

Следовательно, список доступа будет следующим:

```
Router_A(config)# access-list 11
  permit 192.168.30.11 0.0.0.0
Router_A(config)# access-list 11
  permit 192.168.20.0 0.0.0.255
Router_A(config)# int f0/0
Router_A(config-if)# ip access-group 11 out
```

Согласно созданной конфигурации ко всем исходящим из маршрутизатора пакетам через интерфейс f0/0 будет применяться список доступа:

– **permit 192.168.30.11** — разрешение узлу с инвертированной маской WildCard **0.0.0.0**,

– **permit 192.168.20.0** — разрешение сети с инвертированной маской WildCard **0.0.0.255**,
– **deny any** — присутствует неявно в конце списка доступа.

Записи **192.168.30.11 0.0.0.0** полностью соответствует другой вариант — **host 192.168.30.11**, который также предписывает обрабатывать адрес только одного узла.

Пример 4. В Сети рис. 7.6 необходимо установить список доступа, который:
1. блокирует рабочей станции 192.168.20.11 Сети 2 доступ в Сеть1;
2. блокирует рабочей станции 192.168.30.24 Сети 3 доступ в Сеть1.

Для этого создается список доступа:

```
Router_A(config)# access-list 12 deny host 192.168.20.11
Router_A(config)# access-list 12 deny host 192.168.30.24
Router_A(config)# access-list 12 permit any
Router_A(config)# int f0/0
Router_A(config-if)# ip access-group 12 out
```

Данный список блокирует доступ в Сеть 1 только двум рабочим станциям 192.168.20.11 и 192.168.30.24, а всем остальным — доступ разрешен. Если бы отсутствовала третья строка списка доступа, то ни одна станция из других сетей не могла бы попасть в Сеть 1.

7.3.2. Конфигурирование расширенных списков доступа

В отличие от стандартных списков доступа, где в качестве критерия фильтрации только один параметр — адрес источника, **расширенные списки используют несколько параметров**:
– адрес источника,
– адрес назначения,
– протокол,
– порт.

Формат команды создания расширенного списка доступа следующий:

```
Router(config)# access-list {номер} {permit или deny}
  {протокол} {адрес источника} {адрес назначения} {порт}
```

В поле протокола задается имя или номер (0–255) протокола сети Интернет. Наиболее часто используются протоколы IP, TCP, UDP, OSPF, RIP и др. Поле порта используется либо для задания номера (0–65535), либо — имени портов, например, FTP или Telnet.

Формат команды привязки списка доступа к интерфейсу аналогичен команде стандартного списка:

```
Router(config-if)# {протокол} access-group
  {номер} {in или out}
```

Пример 5. В сети (рис. 7.6) необходимо:

1. разрешить одной рабочей станции 192.168.30.11 Сети 3 доступ к серверу с адресом 192.168.10.25 Сети 1 с адресом порта 8080;
2. разрешить всем рабочим станциям Сети 2 с адресом 192.168.20.0 доступ к тому же серверу;
3. разрешить всем рабочим станциям доступ ко всем Web-серверам Сети

Для этого создается список доступа:

```
Router_A(config)# access-list 110 permit tcp
  host 192.168.30.11 host 192.168.10.25 eq 8080
Router_A(config)# access-list 110 permit tcp
  192.168.20.11 0.0.0.255 host 192.168.10.25 eq 8080
Router_A(config)# access-list 110 permit
  tcp any any eq WWW
Router_A(config)# int f0/0
Router_A(config-if)# ip access-group 110 out
```

Запись **any** (все) эквивалентна записи **0.0.0.0 255.255.255.255**, т.е. ни один бит адреса не должен анализироваться. Следовательно, в третьем условии Примера 4 записано требование, исключить фильтрацию по адресу источника и адресу назначения, т.е. запись **permit tcp any any** означает «разрешить доступ всем сегментам tcp ко всем узлам сети». Единственный критерий фильтрации — это **порт eq WWW**.

Запись **eq** означает требование анализа пакетов только с данным номером порта назначения. Вместо нее могла быть другая запись, например, **neq**, означающая требование анализа пакетов с другими номерами, за исключением данного. Запись **range** означает требование анализа пакетов с номерами портов в указанном диапазоне.

Пример 6. Необходимо в сети (рис. 7.6) создать список доступа, чтобы:

1. блокировать рабочей станции 192.168.20.11 Сети 2 доступ по telnet в Сеть 1, но оставить доступ для другого сервиса;
2. блокировать рабочей станции 192.168.30.24 Сети 3 доступ по telnet в Сеть 1, но оставить доступ для другого сервиса;

Для этого создается список доступа:

```
Router_A(config)# access-list 115 deny
  tcp host 192.168.20.11 192.168.10.0 0.0.0.255 eq telnet
Router_A(config)# access-list 115 deny
  tcp host 192.168.30.24 192.168.10.0 0.0.0.255 eq telnet
Router_A(config)# access-list 115 permit ip any any
Router_A(config)# int f0/0
Router_A(config-if)# ip access-group 115 out
```

Удаление списков доступа производится с использованием отрицания **no**. Например, удаление списка доступа из предыдущего примера производится по команде:

```
Router_A(config)# no access-list 115
```

7.3.3. Именованные списки доступа

Именованные списки доступа позволяют за счет введения имени списка сократить затем объем записи при конфигурировании. Кроме того, снимаются ограничения в 99 стандартных и 100 номеров расширенных списков, поскольку имен можно придумать много. Именованный список доступа с именем **spisok** для вышеприведенного примера 4 будет выглядеть следующим образом:

```
Router_A(config)# access-list extended spisok
Router_A(config-ext-nac1)# permit tcp host
  192.168.30.11 host 192.168.10.25 eq 8080
Router_A(config-ext-nac1)# permit tcp
  192.168.20.11 0.0.0.255 host 192.168.10.25 eq 8080
Router_A(config-ext-nac1)# permit tcp any any eq WWW
Router_A(config-ext-nac1)# exit
Router_A(config)# int f0/0
Router_A(config-if)# ip access-group spisok out
```

7.3.4. Контроль списков доступа

Контроль списков доступа производится по командам **show**. Например, контроль любых списков доступа производится по команде:

```
Router_A# show access-list

Extended IP access list 110
permit tcp host 192.168.30.11 host
  192.168.10.25 eq 8080 (34 matches)
permit tcp 192.168.20.11 0.0.0.255 host
  192.168.10.25 eq 8080 (11 matches)
permit tcp any any eq WWW (29 matches)
```

Контроль IP-списков доступа производится по команде:

```
Router_A# show ip access-list
```

Списки доступа, установленные на интерфейсы, можно посмотреть по команде **show ip interface**, а также **show running-config**.

7.4. Анализ MAC-адресов при сетевой фильтрации

Коммутаторы имеют заданную при изготовлении конфигурацию по умолчанию, поэтому коммутатор может работать по умолчанию без изменения базовой IP-конфигурации. Однако эта конфигурация редко удовлетворяет потребности администраторов сети при создании различных виртуальных локальных сетей (VLAN), при формировании информационной безопасности. Коммутаторы могут конфигурироваться и управляться из командной строки интерфейса (command-line

interface — **CLI**), аналогично конфигурированию маршрутизатора. Устройства сети могут также конфигурироваться и управляться через базовый web интерфейс и browser.

7.4.1. Адресация коммутаторов, конфигурирование интерфейсов

Для управления широко распространенным коммутатором Catalist 2950 и более поздними образцами введен **виртуальный интерфейс vlan 1**, на который устанавливается IP-конфигурация; на указанный интерфейс задается IP-адрес, маска сети или подсети, адрес шлюза по умолчанию:

```
Switch2950(config)# interface VLAN1
Switch2950(config-if)# ip address 192.168.1.2 255.255.255.0
Switch2950(config-if)# ip default-gateway 192.168.1.1
```

Чтобы изменить IP-адрес и заданный по умолчанию шлюз, можно либо ввести новый адрес, либо удалить информацию командами глобальной конфигурации **no ip address** или **no ip default-gateway**.

Для верификации конфигурации используется команда **show interface vlan1** в привилегированном режиме:

```
Switch# show interface vlan1
```

Конфигурация коммутатора хранится в NVRAM, также как маршрутизатора. Текущую конфигурацию можно посмотреть по команде **show running-config**.

Для сохранения текущей конфигурации в **NVRAM** администратор может воспользоваться командой **copy running-config startup-config:**

```
Switch# copy run start
```

На коммутаторах Catalist 2950 дуплексный режим и скорость передачи установлены по умолчанию. Однако они могут быть установлены и вручную администратором, если до того дуплексный режим по каким-либо причинам был отменен:

```
Switch(config)# interface FastEthernet0/2
Switch(config-if)# duplex full
Switch(config-if)# speed 100
```

7.4.2. Управление таблицей коммутации

Коммутаторы изучают MAC-адрес источника кадра, полученного на входной интерфейс, и регистрирует его в таблице коммутации. При этом реализуется **динамический режим** формирования таблицы. Кадры, которые имеют MAC-адрес назначения, зарегистрированный в таблице коммутации, могут переключаться только на соответствующий интерфейс без использования широковещательной передачи на все порты. Если в течение заданного времени с какого либо узла

нет передачи кадров, то запись такого узла удаляется из таблицы (см. раздел 5.4). Не дожидаясь истечения заданного времени, администратор может вручную произвести очистку динамически созданных адресов путем использования команды **clear mac-address-table** в привилегированном режиме.

Таблица коммутации (таблица MAC-адресов) может формироваться, изменяться и дополняться **в статическом режиме администратором**, при этом повышается безопасность сети. Чтобы сконфигурировать статическую запись таблицы, т.е. указать, что узел с некоторым MAC-адресом присоединен к определенному порту коммутатора, используется следующая команда:

```
Switch(config)# mac-address-table static
  <MAC-адрес узла> vlan <имя vlan>
  interface FastEthernet <номер>
```

Ниже приведен пример конфигурирования коммутатора `Switch_A`, на котором уже были динамически сформированы три строки таблицы с интерфейсами FA0/7, FA0/8 и FA0/9, отображаемые по команде:

```
Switch_A>sh mac-address-table
Mac Address Table
-------------------------------------------
Vlan  Mac Address  Type  Ports
----  -----------  --------  -----
2  0060.2f2e.9907  DYNAMIC  Fa0/7
3  0060.2f2e.9908  DYNAMIC  Fa0/8
4  0060.2f2e.9909  DYNAMIC  Fa0/9
Switch_A>
```

Затем администратором статически конфигурируется новая запись:

```
Switch-A(config)# mac-address-table static
  0030.A3E9.6623 vlan 2 Interface FastEthernet 0/2,
```

которая отображается в таблице коммутации (Type — STATIC):

```
Switch-A#sh mac-address-table
Mac Address Table
-------------------------------------------
Vlan  Mac Address  Type  Ports
----  -----------  --------  -----
2  0030.a3e9.6623  STATIC  Fa0/2
2  0060.2f2e.9907  DYNAMIC  Fa0/7
3  0060.2f2e.9908  DYNAMIC  Fa0/8
4  0060.2f2e.9909  DYNAMIC  Fa0/9
```

Подобную информацию можно также увидеть по команде **sh run**:

```
Switch_A#sh run
...
mac-address-table static
  0030.a3e9.6623 vlan 2 interface FastEthernet0/2
```

Чтобы удалить созданные статически MAC-адреса, нужно использовать отрицательную форму команды:

```
Switch(config)# no mac-address-table static
  <MAC-адрес узла> interface FastEthernet <номер>
  vlan <номер>
```

7.4.3. Конфигурирование безопасности на коммутаторе

Порты коммутатора доступны через структурированную кабельную систему. Любой может включиться в один из портов, что является потенциальным пунктом входа в сеть неправомочным пользователем. При этом злоумышленник может сконфигурировать коммутатор так, чтобы он функционировал как концентратор, что позволяет проанализировать весь трафик сети, проходящий через коммутатор. Поэтому коммутаторы должны обеспечивать **безопасность портов** (port security).

Статическое конфигурирование администратором MAC-адресов обеспечивает безопасность путем жесткой привязки адреса к интерфейсу, однако это достаточно сложно. Для обеспечения динамического режима безопасности используется ряд команд конфигурирования коммутатора. Например, динамический режим обеспечения безопасности на интерфейсе Fast Ethernet 0/7 конфигурируется следующей последовательностью команд:

```
Switch_A(config-if)# int f0/7
Switch_A(config-if)# switchport port-security
```

или последовательностью, используемой в виртуальных локальных сетях

```
Switch_A(config)# int f0/7
Switch_A(config-if)# switchport mode access
Switch_A(config-if)# switchport port-security
```

После ввода указанной последовательности команд таблица коммутации приобретает следующий вид:

```
Switch-A#sh mac-address-table
Mac Address Table
-------------------------------------------
Vlan  Mac Address  Type  Ports
----  -----------  --------  -----
2  0030.a3e9.6623  STATIC  Fa0/2
2  0060.2f2e.9907  STATIC  Fa0/7
3  0060.2f2e.9908  DYNAMIC  Fa0/8
4  0060.2f2e.9909  DYNAMIC  Fa0/9
```

То есть привязка адреса к интерфейсу реализуется автоматически (динамически), но режим формирования записи становится статическим, т.е. запись динамически не удаляется по истечению времени таймера.

С целью повышения безопасности ограничивают число MAC-адресов конечных узлов, которым разрешено присоединяться к данному интерфейсу коммутатора. Например, **число MAC-адресов на порт может быть ограничено до 1**. В этом случае первый адрес, динамически изученный коммутатором, считается безопасным адресом. Всем узлам с другими MAC-адресами доступ будет запрещен. Ограничение число MAC-адресов на порт до 1 реализуется следующей последовательностью команд:

```
Switch_A(config)#int fa 0/7
Switch_A(config-if)#switchport port-security max 1
```

Верификация режима **port security** конкретного интерфейса обеспечивается командой **show port security**:

```
Switch-A#sh port-security int f0/7

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  : 0
Sticky MAC Addresses  : 0
Last Source Address:Vlan  : 0060.2F2E.9907:2
Security Violation Count  : 0
```

Третья строка распечатки показывает **режим реагирования** системы на нарушения безопасности, который по умолчанию установлен в состояние «**Выключение**» (**Shutdown**). Нарушение безопасности происходит, когда станция, чей MAC-адрес отсутствует в таблице коммутации, пытается получить доступ к интерфейсу. При этом порт немедленно **выключается** и формируется сообщение о нарушении безопасности. Существуют еще два режима реагирования на нарушения безопасности: режим защиты (Protect) и режим ограничения (Restrict). В этих режимах пакеты с неизвестными исходящими MAC-адресами уничтожаются. При этом в режиме ограничения формируется уведомление, а в режиме защиты — не формируется. Установить режим «Выключение» можно по команде:

```
Switch_A(config-if)#switchport port-security
  violation shutdown
```

По умолчанию все порты коммутатора находятся во включенном состоянии. Для повышения безопасности рекомендуется выключать все неиспользуемые порты коммутатора по команде **Shutdown**. Ниже приведен пример фрагмента распечатки команды **sh run**, где показано, что интерфейс FastEthernet0/10 выключен.

```
Switch_A#sh run
!
interface FastEthernet0/7
  switchport access vlan 2
  switchport mode access
  switchport port-security
!
interface FastEthernet0/8
  switchport access vlan 3
  switchport mode access
!
interface FastEthernet0/9
  switchport access vlan 4
  switchport mode access
!
interface FastEthernet0/10
  shutdown
!
interface FastEthernet0/11
!
```

Поскольку для включения портов требуется знание паролей входа через консольную или виртуальную линию, а также пароль входа в привилегированный режим, то выключение неиспользуемых портов коммутатора повышает безопасность сетевого устройства.

Выключение режима безопасности **port security** обеспечивается формой **no** команды, по которой режим вводился.

Таким образом, введение режима безопасности (**switchport port-security**) и ограничение числа MAC-адресов конечных узлов на порт коммутатора, которым разрешено присоединяться к данному интерфейсу, позволяет эффективно экранировать защищаемую сеть от неавторизованных пользователей.

7.5. Виртуальные локальные сети

7.5.1. Общие сведения о виртуальных локальных сетях

Безопасность телекоммуникационных сетей во многом определяется размерами широковещательных доменов, внутри которых может происходить несанкционированный доступ к конфиденциальной информации. В традиционных сетях деление на широковещательные домены реализует маршрутизатор.

Виртуальные сети созданы, чтобы реализовать сегментацию сети на коммутаторах. Таким образом, создание виртуальных локальных сетей (Virtual Local Area Networks — **VLAN**), которые представляют собой логическое объединение групп станций сети (рис. 7.7), является

одним из основных методов защиты информации в сетях на коммутаторах.

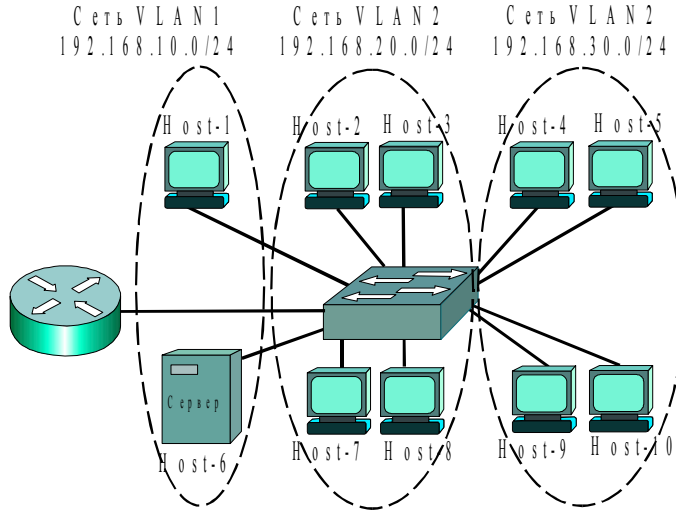


Рис. 7.7. Виртуальные локальные сети VLAN

Обычно VLAN группируются по функциональным особенностям работы, независимо от физического местоположения пользователей. Обмен данными происходит только между устройствами, находящимися в одной VLAN. Обмен данными между различными VLAN производится только через маршрутизаторы.

Рабочая станция в виртуальной сети, например, Host-1 в сети VLAN1 (рис. 7.7), ограничена общением с сервером в той же самой VLAN1. Виртуальные сети логически сегментируют всю сеть на широковещательные домены так, чтобы пакеты переключались только между портами, которые назначены на ту же самую VLAN (приписаны к одной VLAN). Каждая сеть VLAN состоит из узлов, объединенных единственным широковещательным доменом, образованным приписанными к виртуальной сети портами коммутатора.

Поскольку каждая виртуальная сеть представляет широковещательный домен, то маршрутизаторы в топологии сетей VLAN (рис. 7.7) обеспечивают фильтрацию широковещательных передач, безопасность, управление трафиком и связь между VLAN. Коммутаторы не обеспечивают трафик между VLAN, поскольку это нарушает целостность широковещательного домена VLAN. **Трафик между VLAN** обеспечивается маршрутизацией, т.е. **общение между узлами разных**

виртуальных сетей происходит только через маршрутизатор.

Для нормального функционирования виртуальных сетей необходимо на коммутаторе сконфигурировать все виртуальные локальные сети и приписать порты коммутатора к соответствующей сети VLAN. Если кадр должен пройти через коммутатор и MAC-адрес назначения известен, то коммутатор только продвигает кадр к соответствующему выходному порту. Если MAC-адрес неизвестен, то происходит широковещательная передача во все порты широковещательного домена, т.е. внутри виртуальной сети VLAN, кроме исходного порта, откуда кадр был получен. **Широковещательные передачи снижают безопасность информации**.

Управление виртуальными сетями VLAN по умолчанию реализуется через первую сеть VLAN1 и сводится к управлению портами коммутатора. Сеть VLAN1 получила название **сеть по умолчанию** (**default VLAN**). По крайней мере, один порт должен быть в VLAN 1, чтобы управлять коммутатором. Все другие порты на коммутаторе могут быть назначены другим сетям VLAN. Поскольку данная информация известна всем, то хакеры пытаются атаковать, в первую очередь, именно эту сеть. Поэтому на практике администраторы изменяют номер сети по умолчанию, например, на номер VLAN 101.

Каждой виртуальной сети при конфигурировании должен быть назначен IP-адрес сети или подсети с соответствующей маской, для того чтобы виртуальные сети могли общаться между собой. Например, VLAN1 (рис. 7.7) может иметь адрес 192.168.10.0/24, VLAN2 — адрес 192.168.20.0/24, VLAN3 — адрес 192.168.30.0/24. Каждому хосту необходимо задать IP-адрес из диапазона адресов соответствующей виртуальной сети, например, host-1 — адрес 192.168.10.1, host-2 – адрес 192.168.20.1, host-3 — адрес 192.168.20.2, host-7 — адрес 192.168.20.3, host-10 — адрес 192.168.30.4.

Идентификаторы виртуальных сетей (VLAN1, VLAN2, VLAN3 и т.д.) могут назначаться из нормального диапазона 1–1005, в котором номера 1002–1005 зарезервированы для виртуальных сетей технологий Token Ring и FDDI. Существует также расширенный диапазон идентификаторов 1006–4094. Однако для облегчения управления рекомендуется, чтобы сетей VLAN было не более 255 и сети не расширялись вне Уровня 2 коммутатора.

Таким образом, сеть VLAN является широковещательным доменом, созданным одним или более коммутаторами. На рис. 7.8, три виртуальных сети VLAN созданы одним маршрутизатором и тремя коммутаторами. При этом существуют три отдельных широковещательных домена (сеть VLAN1, сеть VLAN2, сеть VLAN3). Маршрутизатор управляет трафиком между сетями VLAN, используя маршрутизацию Уровня 3.

Если рабочая станция сети VLAN1 захочет послать кадр рабочей станции в той же самой VLAN1, адресом назначения кадра будет MAC-

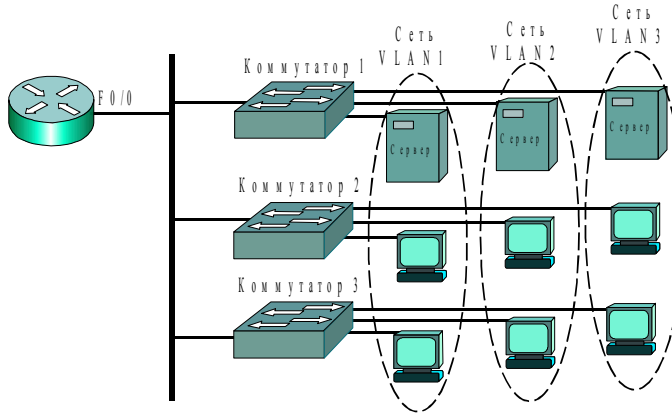


Рис. 7.8. Три виртуальных сети VLAN

адрес рабочей станции назначения. Если же рабочая станция сети VLAN1 захочет переслать кадр рабочей станции сети VLAN2, кадры будут переданы на MAC-адрес интерфейса F0/0 маршрутизатора. То есть, маршрутизация производится через IP-адрес интерфейса F0/0 маршрутизатора виртуальной сети VLAN1.

Для выполнения своих функций в виртуальных сетях коммутатор должен поддерживать таблицы коммутации (продвижения) для каждой VLAN. Для продвижения кадров производится поиск адреса в таблице только данной VLAN. Если адрес источника ранее не был известен, то при получении кадра коммутатор динамически добавляет этот адрес в таблицу.

При построении сети на нескольких коммутаторах необходимо выделить дополнительные порты для объединения портов разных коммутаторов, приписанных к одноименным виртуальным сетям (рис. 7.9). Дополнительных пар портов двух коммутаторов должно быть выделено столько, сколько создано сетей VLAN.

Поскольку кадры данных могут быть получены коммутатором от любого устройства, присоединенного к любой виртуальной сети, то при обмене данными между коммутаторами в заголовок кадра добавляется **уникальный идентификатор** кадра — **тег** (tag) виртуальной сети, который определяет VLAN каждого пакета. **Стандарт IEEE 802.1Q** предусматривает введение **поля меток** в заголовок кадра, содержащее два байта (рис. 7.10).

Из них 12 двоичных разрядов используются для адресации, что позволяет помечать до 4096 виртуальных сетей и соответствует нормальному и расширенному диапазону идентификаторов VLAN. Еще

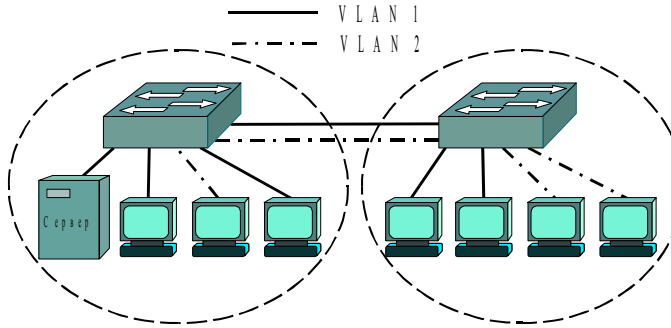


Рис. 7.9. Объединение виртуальных сетей двух коммутаторов

3 бита	1 бит	12 бит
Приоритет	CFI	VLAN ID

Рис. 7.10. Формат тега виртуальной сети

три разряда этого поля позволяют задавать 8 уровней приоритета передаваемых сообщений, т.е. позволяют обеспечивать качество (QoS) передаваемых данных. Наивысший приоритет уровня 7 имеют кадры управления сетью, уровень 6 — кадры передачи голосового трафика, 5 — передача видео. Остальные уровни обеспечивают передачу данных с разным приоритетом. Единичное значение поля CFI показывает, что виртуальная сеть является Token Ring.

Пакет отправляется коммутатором или маршрутизатором, базируясь на идентификаторе VLAN и MAC-адресе. После достижения сети назначения идентификатор VLAN (tag) удаляется из пакета коммутатором, а пакет отправляется присоединенному устройству. **Маркировка пакета** (Packet tagging) обеспечивает механизм управления потоком данных.

7.5.2. Транковые соединения

Согласно принципу, представленному на рис.7.9, в виртуальных локальных сетях для соединения нескольких коммутаторов между собой задействуют несколько физических портов по числу виртуальных локальных сетей. Совокупность физических каналов между двумя устройствами (рис. 7.11) может быть заменена **одним агрегированным логическим каналом** (рис. 7.12), получившим название **транк** (Trunk).

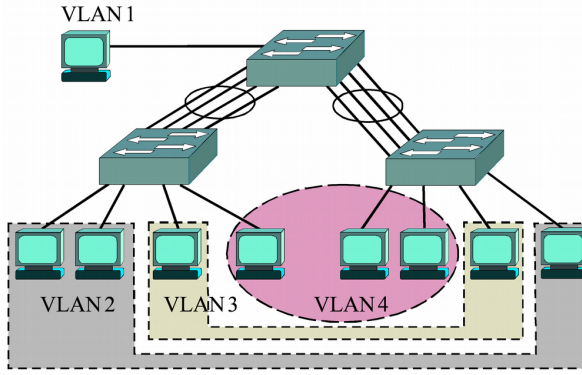


Рис. 7.11. Несколько физических портов по числу виртуальных локальных сетей

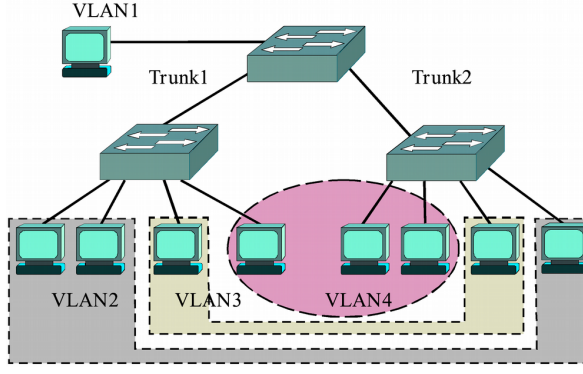


Рис. 7.12. Транковые соединения коммутаторов

Транковые соединения используются и для подключения маршрутизатора к коммутатору. При этом на интерфейсе маршрутизатора формируются несколько субинтерфейсов (по количеству виртуальных сетей). Пропускная способность агрегированного логического канала равна сумме пропускных способностей отдельных подканалов. Транки используют и для подключения высокоскоростных серверов.

На практике используются статические и динамические VLAN. Динамические VLAN создаются через программное обеспечение управ-

ления сети. Однако динамические VLAN широко не используется.

Наибольшее распространение получили статические VLAN. Входящие в сеть устройства автоматически становятся членами VLAN порта, к которому они присоединены. Для статического конфигурирования VLAN используется интерфейс командной линии CLI.

7.5.3. Конфигурирование виртуальных локальных сетей

Конфигурационный файл в виде базы данных **vlan.dat**, хранится во флэш-памяти коммутатора. Каждая VLAN должна иметь уникальный адрес Уровня 3 или выделенный ей адрес подсети. Это позволяет маршрутизаторам переключать пакеты между виртуальными локальными сетями.

Статическое конфигурирование виртуальных сетей сводится к назначению портов коммутатора на каждую виртуальную локальную сеть VLAN, что может непосредственно конфигурироваться на коммутаторе через использование командной строки CLI. Таким образом, при статическом конфигурировании каждый порт приписывается к какой-то виртуальной сети. Статически сконфигурированные порты поддерживают назначенную конфигурацию до тех пор, пока не будут изменены вручную. **Пользователи** подключены к портам коммутатора на **уровне доступа** (access layer). **Маркировка** (Frame tagging) используется, чтобы обмениваться информацией внутри сетей VLAN между коммутаторами.

По умолчанию управляющей сетью является первая сеть VLAN 1, однако ей может быть назначен другой номер, причем, сеть VLAN 1 — будет Ethernet сетью, и ей принадлежит IP-адрес коммутатора.

Ниже рассмотрено конфигурирование коммутатора для виртуальной локальной сети (рис. 7.13).

Примеры конфигурирования даны для коммутаторов серии 2950 и последующих модификаций.

Состояние виртуальных сетей и интерфейсов коммутатора Catalyst серии 2950-24 с именем `Sw_A` можно посмотреть по следующей команде:

```
Sw_A#sh 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
1002 fddi-default  active
1003 token-ring-default  active
```

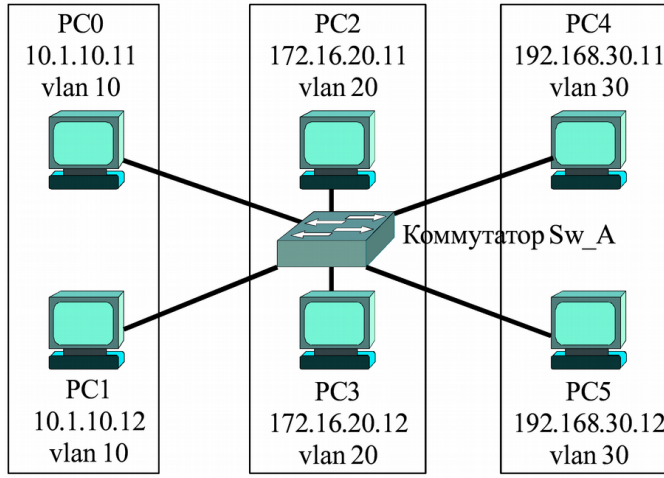


Рис. 7.13. Виртуальная локальная сеть

```
1004 fddinet-default  active
1005 trnet-default  active
```

Из распечатки команды `Sw_A#sh vlan brief` следует, что все 24 интерфейса FastEthernet приписаны к сети по умолчанию VLAN 1, других активных виртуальных сетей нет, за исключением 1002–1005, зарезервированных для сетей token-ring и fddi.

Создание виртуальных сетей может производиться двумя способами:

1. в режиме глобального конфигурирования;
2. по команде **vlan database** из привилегированного режима конфигурирования.

Примеры конфигурирования трех виртуальных локальных сетей (рис. 7.13) vlan 10, vlan 20, vlan 30 приведены ниже.

При первом способе используются следующие команды:

```
Sw-A(config)# vlan 10
Sw-A(config-vlan)# vlan 20
Sw-A(config-vlan)# vlan 30
```

По второму способу:

```
Sw-A# vlan database
Sw-A(vlan)# vlan 10
Sw-A(vlan)# vlan 20
Sw-A(vlan)# vlan 30
```

Обычно рекомендуют использовать первый способ создания виртуальных локальных сетей.

После задания виртуальных сетей vlan 10, vlan 20, vlan 30 они становятся активными, что можно посмотреть по команде **sh vlan brief**:

```
Sw-A# sh 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
10  VLAN0010  active
20  VLAN0020  active
30  VLAN0030  active
1002 fddi-default  active
1003 token-ring-default  active
1004 fddinet-default  active
1005 trnet-default  active
```

При желании можно также сформировать название VLAN по команде **vlan номер name имя**, например:

```
Switch2950(config-vlan)# vlan 30 name VLAN30
```

или

```
Switch2950(vlan)# vlan 3 name VLAN3
```

Указанные операции не являются обязательными, они служат только для удобства чтения распечаток.

На следующем этапе необходимо назначить виртуальные сети на определенные интерфейсы (приписать интерфейсы к созданным виртуальным сетям), используя пару команд **switchport mode access**, **switchport access vlan №**. Ниже приведен пример указанных операций для сети рис. 7.13.

```
Sw-A(config)# int f0/1
Sw-A(config-if)# switchport mode access
Sw-A(config-if)# switchport access vlan 10
Sw-A(config-if)# int f0/2
Sw-A(config-if)# switchport mode access
Sw-A(config-if)# switchport access vlan 10
Sw-A(config-if)# int f0/3
Sw-A(config-if)# switchport mode access
Sw-A(config-if)# switchport access vlan 20
Sw-A(config-if)# int f0/4
Sw-A(config-if)# switchport mode access
```

```
Sw-A(config-if)# switchport access vlan 20
Sw-A(config-if)# int f0/5
Sw-A(config-if)# switchport mode access
Sw-A(config-if)# switchport access vlan 30
Sw-A(config-if)# int f0/6
Sw-A(config-if)# switchport mode access
Sw-A(config-if)# switchport access vlan 30
```

Если при конфигурировании нескольких портов режим не изменяется, то команда **switchport mode access** может использоваться один раз для первого интерфейса. Верификацию полученной конфигурации можно произвести с помощью команд **show vlan** или **show vlan brief**, например:

```
Sw-A#sh vlan
VLAN Name  Status    Ports
---- ----------------  --------
-----------------------------
1  default      active      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
10  VLAN0010  active    Fa0/1, Fa0/2,
20  VLAN0020  active    Fa0/3, Fa0/4,
30  VLAN0030  active    Fa0/5, Fa0/6,
1002 fddi-default  active
1003 token-ring-default  active
1004 fddinet-default  active
1005 trnet-default  active
VLAN Type  SAID  MTU  Parent RingNo BridgeNo
  Stp BrdgMode Trans1 Trans2
---- ----- ------  ----  ----- ------
-------  ---- --------  ------ -----
1  enet  100001  1500  -  -  -  -  -  0  0
10  enet  100010  1500  -  -  -  -  -  0  0
20  enet  100020  1500  -  -  -  -  -  0  0
30  enet  100030  1500  -  -  -  -  -  0  0
1002 enet  101002  1500  -  -  -  -  -  0  0
1003 enet  101003  1500  -  -  -  -  -  0  0
1004 enet  101004  1500  -  -  -  -  -  0  0
1005 enet  101005  1500  -  -  -  -  -  0  0
```

Из распечатки следует, что команда **show vlan** дает больше информации, чем **show vlan brief**.

Кроме того, конфигурацию конкретной виртуальной сети, например VLAN2, можно также просмотреть с помощью команд **show vlan id 2** или по имени **show vlan name VLAN 2**, если имя задано.

Конфигурационный файл коммутатора должен быть скопирован в энергонезависимую память коммутатора по команде

```
Sw-A# copy running-config startup-config
```

Он может быть также скопирован на сервер TFTP с помощью команды **copy running-config tftp**. Параметры конфигурации можно посмотреть с помощью команд **show running-config** или **show vlan**.

Удаление виртуальной сети, например vlan 10, выполняется с помощью формы **no** команды:

```
Sw-A(config)# no vlan 10
```

или

```
Switch# vlan database
Switch(vlan)# no vlan 10
```

Когда виртуальная локальная сеть удалена, все порты, приписанные к этой VLAN, становятся бездействующими. Однако порты останутся связанными с удаленной виртуальной сетью VLAN пока не будут приписаны к другой виртуальной сети или не будет восстановлена прежняя.

Для того, чтобы отменить неверное назначение интерфейса на виртуальную сеть, например, ошибочное назначение виртуальной сети vlan 20 на интерфейс F0/2, используется команда:

```
Sw-A(config)# int f0/2
Sw-A(config-if)# no switchport access vlan
```

Также можно было бы просто приписать интерфейс f0/2 к другой виртуальной сети, например, к vlan 10:

```
Sw-A(config)# int f0/2
Sw-A(config-if)# switchport mode access
Sw-A(config-if)# switch access vlan 10
```

В рассматриваемом примере на конечных узлах (хостах) сети рис. 7.13 установлена следующая конфигурация (табл. 7.3).

Таблица 7.3

Конфигурация конечных узлов виртуальных локальных сетей

VLAN №	Узел	Адрес узла	Маска	Шлюз
Vlan 10	PC0 PC1	10.1.10.11 10.1.10.12	255.255.255.0	10.1.10.1
Vlan 20	PC2 PC3	172.16.20.11 172.16.20.12	255.255.255.0	172.16.20.1
Vlan 30	PC4 PC5	192.168.30.11 192.168.30.12	255.255.255.0	192.168.30.1

Таким образом, каждая виртуальная локальная сеть имеет свой IP-адрес.

Проверка работоспособности сети производится по командам ping, (tracert). Она показывает, что, например, PC0 имеет соединение с PC1:

```
PC0>ping 10.1.10.11
Pinging 10.1.10.11 with 32 bytes of data:
Reply from 10.1.10.11: bytes=32 time=82ms TTL=128
Reply from 10.1.10.11: bytes=32 time=80ms TTL=128
Reply from 10.1.10.11: bytes=32 time=73ms TTL=128
Reply from 10.1.10.11: bytes=32 time=70ms TTL=128
Ping statistics for 10.1.10.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 70ms, Maximum = 82ms, Average = 76ms
```

но не может обмениваться сообщениями с узлами других VLAN:

```
PC0>ping 172.16.20.11
Pinging 172.16.20.11 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 172.16.20.11:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Если к сети присоединить дополнительный узел PC6, адрес которого 192.168.30.101, т.е. адрес его сети совпадает с адресом сети vlan 30, но узел PC6 не приписан ни к одной из виртуальных сетей, то он не сможет реализовать соединения с узлами существующих виртуальных сетей.

7.5.4. Маршрутизация между виртуальными локальными сетями

Поскольку каждая виртуальная локальная сеть представляет собой широковещательный домен, т.е. сеть со своим IP-адресом, то для связи между сетями необходима маршрутизация Уровня 3. Поэтому к коммутатору необходимо присоединить маршрутизатор (рис. 7.14).

Для соединения с маршрутизатором в схеме дополнительно задействованы три интерфейса коммутатора `Sw_A`: F0/11, F0/12, Ff0/13. При этом порт F0/11 приписан к сети vlan 10, порт F0/12 — к vlan 20, порт F0/13 — к vlan 30.

```
Sw_A(config)# int f0/11
Sw_A(config-if)# switchport access vlan 10
Sw_A(config-if)# int f0/12
Sw_A(config-if)# switchport access vlan 20
Sw_A(config-if)# int f0/13
Sw_A(config-if)# switchport access vlan 30
```

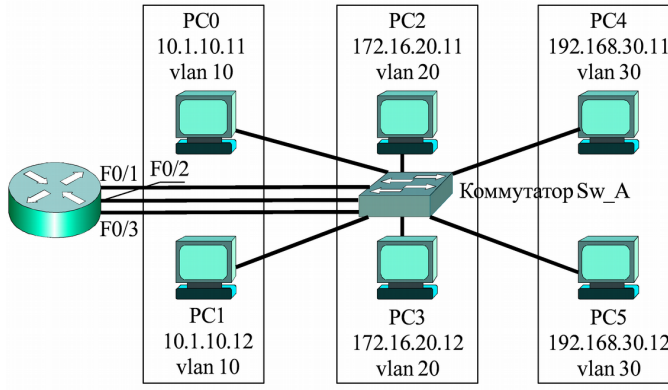


Рис. 7.14. Связь между сетями через маршрутизатор

На маршрутизаторе используются три интерфейса F0/1, F0/2, F0/3 (по числу виртуальных сетей), которые сконфигурированы следующим образом:

```
Router> ena
Router# conf t
Router(config)# int f0/1
Router(config-if)# ip add 10.1.10.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# int f0/2
Router(config-if)# ip add 172.16.20.1 255.255.255.0
Router(config-if)# no shut
Router(config)# int f0/3
Router(config-if)# ip add 192.168.30.1 255.255.255.0
Router(config-if)# no shut
```

По команде **sh ip route** можно посмотреть таблицу маршрутизации:

```
Router#sh ip route
Gateway of last resort is not set
  10.0.0.0/24 is subnetted, 1 subnets
  C 10.1.10.0 is directly connected, FastEthernet0/1
    172.16.0.0/24 is subnetted, 1 subnets
  C 172.16.20.0 is directly connected, FastEthernet0/2
  C 192.168.30.0/24 is directly connected, FastEthernet0/3
```

Из таблицы маршрутизации следует, что все три сети (10.1.10.0, 172.16.20.0, 192.168.30.0) являются непосредственно присоединенными и, следовательно, могут обеспечивать маршрутизацию между сетями.

«Прозвонка» с узла 10.1.10.11 узлов сетей 172.16.20.0, 192.168.30.0 дает положительный результат.

Защита межсетевых соединений через маршрутизатор может быть реализована с помощью сетевых фильтров (списков доступа).

Недостатком такого метода организации межсетевых соединений является необходимость использования дополнительных интерфейсов коммутатора и маршрутизатора, число которых равно количеству виртуальных сетей. От этого недостатка свободно **транковое** соединение, когда совокупность физических каналов между двумя устройствами может быть заменена одним каналом.

7.5.5. Конфигурирование транковых соединений

При транковом соединении коммутатора и маршрутизатора три физических канала между ними (рис. 7.14) заменяются одним агрегированным каналом (рис. 7.15).

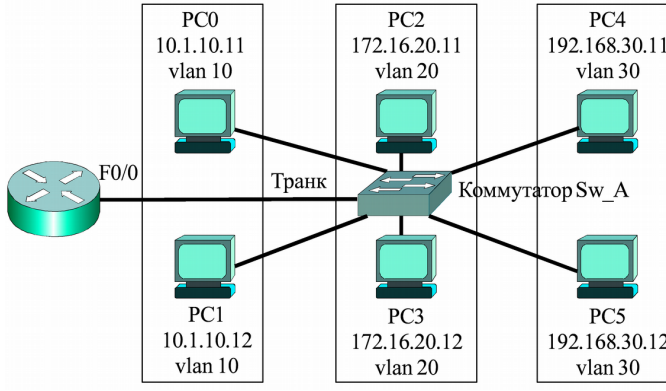


Рис. 7.15. Транковое соединение коммутатора и маршрутизатора

Для создания транкового соединения на коммутаторе задействован интерфейс F0/10, а на маршрутизаторе — F0/0.

Конфигурирование коммутатора будет следующим:

```
Sw_A> ena
Sw_A# conf t
Sw_A(config)# vlan 10
Sw_A(config-vlan)# vlan 20
Sw_A(config-vlan)# vlan 30
Sw_A(config-vlan)# int f0/1
Sw_A(config-if)# switchport mode access
```

```
Sw_A(config-if)# switchport access vlan 10
Sw_A(config-if)# int f0/4
Sw_A(config-if)# switchport access vlan 10
Sw_A(config-if)# int f0/2
Sw_A(config-if)# switchport access vlan 20
Sw_A(config-if)# int f0/5
Sw_A(config-if)# switchport access vlan 20
Sw_A(config-if)# int f0/3
Sw_A(config-if)# switchport access vlan 30
Sw_A(config-if)# int f0/6
Sw_A(config-if)# switchport access vlan 30
Sw_A(config-if)# int f0/10
Sw_A(config-if)# switchport mode trunk
Sw_A(config-if)# ^Z
```

По команде **sh int f0/10 switchport** можно посмотреть состояние интерфейса:

```
Sw_A# sh int f0/10 switchport

Name: Fa0/10
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
. . .
Sw_A#
```

Из распечатки следует, что порт F0/10 находится в режиме Trunk.

Конфигурирование маршрутизатора сводится к тому, что на его интерфейсе F0/0 формируются субинтерфейсы F0/0.10, F0/0.20, F0/0.30. На указанных субинтерфейсах задается протокол Dot 1q для виртуальных сетей 10, 20, 30. Последовательность команд необходимо завершить включением интерфейса **no shutdown**.

```
Router> ena
Router# conf t
Router(config-if)# int f0/0.10
Router(config-subif)# encapsulation dot1q 10
Router(config-subif)# ip add 10.1.10.1 255.255.255.0
Router(config-subif)# int f0/0.20
Router(config-subif)# encapsulation dot1q 20
Router(config-subif)# ip add 172.16.20.1 255.255.255.0
Router(config-subif)# int f0/0.30
Router(config-subif)# encapsulation dot1q 30
```

```
Router(config-subif)# ip add 192.168.30.1 255.255.255.0
Router(config-subif)# int f0/0
Router(config-if)# no shut
```

Результат конфигурирования проверяется по команде **sh ip route**:

```
Router# sh ip route
Gateway of last resort is not set
  10.0.0.0/24 is subnetted, 1 subnets
C 10.1.10.0 is directly connected, FastEthernet0/0.10
  172.16.0.0/24 is subnetted, 1 subnets
C 172.16.20.0 is directly connected, FastEthernet0/0.20
C 192.168.30.0/24 is directly connected, FastEthernet0/0.30
  Router#
```

Из таблицы маршрутизации следует, что сети 10.1.10.0, 172.16.20.0, 192.168.30.0 являются непосредственно присоединенными. Поэтому маршрутизатор способен обеспечить маршрутизацию между сетями.

Таким образом, транковое соединение представляет собой одно физическое соединение, внутри которого сформировано несколько логических каналов по числу VLAN.

7.6. Краткие итоги раздела

1. Стандарт ISO/IEC 27002 определяет информационную безопасность как сохранение конфиденциальности, целостности и доступности.
2. Формирование набора паролей необходимо для защиты сетевых элементов (коммутаторов и маршрутизаторов) от несанкционированного доступа не авторизованных пользователей, что может привести к нарушению конфиденциальности, целостности и доступности информации.
3. Для защиты информации широко используются сетевые фильтры или списки доступа (Access Lists — ACL).
4. Списки доступа могут использоваться, чтобы разрешать (permit) или запрещать (deny) продвижение пакетов через маршрутизатор.
5. В списке доступа ACL могут анализироваться адреса источника, адреса назначения, протокол и номера порта верхнего уровня.
6. Списки доступа могут быть определены для каждого установленного на интерфейсе протокола и для каждого направления сетевого трафика (исходящего и входящего).
7. Стандартные списки доступа (Standard access lists) для принятия решения анализируют в IP пакете только адрес источника сообщения.
8. Расширенные списки доступа (Extended access lists) проверяют IP-адрес источника, IP-адрес назначения, поле протокола в заголовке пакета Сетевого уровня и номер порта в заголовке Транспортного уровня.

9. Условие **deny any** (запретить все остальное) есть неявно в конце любого списка доступа.
10. Именованные списки доступа позволяют за счет введения имени списка сократить затем объем записи при конфигурировании.
11. Коммутатором необходимо управлять, для чего задаются IP-адрес, маска, шлюз на интерфейс виртуальной локальной сети vlan 1.
12. Кадры, которые имеют MAC-адрес назначения, зарегистрированный в таблице коммутации, переключаются только на соответствующий интерфейс без использования широковещательной передачи на все порты, что повышает безопасность.
13. Статическое конфигурирование повышает безопасность путем жёсткой привязки MAC-адреса к порту коммутатора. Число MAC-адресов на порт может быть ограничено до одного.
14. Виртуальная локальная сеть VLAN состоит из узлов, объединённых единственным широковещательным доменом, образованным приписанными к виртуальной сети портами коммутатора.
15. Для функционирования VLAN необходимо на коммутаторе сконфигурировать все виртуальные локальные сети и приписать порты коммутатора к соответствующей сети.
16. Трафик между VLAN обеспечивается маршрутизацией, т.е. общение между узлами разных виртуальных сетей происходит только через маршрутизатор.
17. Каждой виртуальной сети при конфигурировании должен быть назначен IP-адрес сети или подсети с соответствующей маской и шлюзом.
18. При обмене данными между коммутаторами в заголовок добавляется уникальный идентификатор кадра — тег (tag) виртуальной сети, который определяет членство VLAN каждого пакета.
19. Совокупность физических каналов между двумя устройствами может быть заменена одним агрегированным каналом, получившим название транк (Trunk).
20. Конфигурирование виртуальных сетей сводится к назначению портов коммутатора на каждую виртуальную локальную сеть VLAN.

7.7. Вопросы по разделу

1. Для чего используются сетевые фильтры или списки доступа?
2. На основании чего формируется запрет или разрешение сетевого трафика через интерфейс маршрутизатора?
3. Какие параметры пакета могут анализироваться в списке доступа?
4. Где устанавливаются списки доступа?
5. Что анализируют стандартные списки доступа?
6. Что анализируют расширенные списки доступа?
7. Какое условие имеется неявно в конце любого списка доступа?
8. Каков формат команды создания стандартного списка доступа?

9. Каков формат команды создания расширенного списка доступа?
10. Каков формат команды привязки списка к интерфейсу?
11. Какие достоинства имеют именованные списки доступа?
12. По каким командам конфигурируется IP-адрес и шлюз коммутатора?
13. По какой команде конфигурируется администратором новая запись в таблицу коммутации?
14. По какой команде производится очистка таблицы коммутации?
15. По какой команде конфигурируется динамический режим обеспечения безопасности на интерфейсе?
16. По какой команде можно посмотреть содержимое таблицы коммутации?
17. Для чего создаются виртуальные локальные сети?
18. Как связываются между собой VLAN и порты коммутатора?
19. Как обеспечивается общение между узлами разных виртуальных сетей?
20. Как обеспечивается управление виртуальными локальными сетями?
21. Можно ли построить VLAN на нескольких коммутаторах?
22. Для чего служит идентификатор кадра (tag)? Где он размещается?
23. Что такое транк? Как он создается на коммутаторе и маршрутизаторе?
24. Какие команды используются для назначения VLAN на интерфейсы?
25. Какие команды используются для создания транковых соединений?
26. Какие команды используются для верификации VLAN?

7.8. Задания для самостоятельной работы

Задание 7.1 — Проведите проверку и отладку сети с использованием команд **show running-config**, **show access-list**, **show ip access-list**, **show ip interface**, **ping**, **traceroute** и **tracert**.

Задание 7.2 — Сконфигурируйте на маршрутизаторе и коммутаторе все известные Вам пароли. Проверьте их действие.

Задание 7.3 — С одного из компьютеров выполните удаленный доступ к коммутатору, к маршрутизатору. Измените их имена. Завершите сеанс удаленного доступа. Проведите верификацию проделанной работы.

Задание 7.4 — Проведите проверку таблицы коммутации. Смоделируйте формирование дополнительные динамических записей в таблице коммутации.

Задание 7.5 — Переведите динамические записи в статические.

Задание 7.6 — Сконфигурируйте две виртуальных локальных сети на двух коммутаторах, используя транковые соединения.

Задание 7.7 — Обеспечьте межсетевое взаимодействие.

Глава 8. Сети следующего поколения

8.1. Определение и суть NGN

Сеть связи следующего поколения (ССП — Next Generation Network, NGN) — концепция построения сетей связи, обеспечивающих предоставление неограниченного набора услуг с гибкими возможностями по их управлению, персонализации и созданию новых услуг за счёт унификации сетевых решений, предполагающая реализацию универсальной транспортной сети с распределённой коммутацией, вынесение функций предоставления услуг в оконечные сетевые узлы и интеграцию с традиционными сетями связи.

В сети NGN предоставляют широкий выбор технологий доступа, поставщиков услуг и самих услуг. Кроме того, в сетях NGN пользователи получают доступ к услугам независимо от местоположения и технического окружения, что позволяет обеспечить единообразие в предоставлении услуг.

8.1.1. Основополагающие характеристики NGN

Архитектура NGN предполагает чёткое разделение между функциями обслуживания и функциями транспортировки, что позволяет предоставлять и развивать как существующие, так и новые услуги вне зависимости от используемой сети и типа доступа.

Сети NGN обладают следующими основными характеристиками:

- пакетная коммутация;
- разделение ресурсов между пропускной способностью каналаносителя, вызовом/сеансом, приложением/услугами;
- разделение между предоставлением услуг и транспортировкой, предоставление открытых интерфейсов;
- поддержка широкого спектра услуг, приложений и механизмов на основе унифицированных блоков обслуживания (включая услуги в режиме реального масштабе времени, в потоковом или автономном режиме, мультимедийные услуги);
- возможности широкополосной передачи со сквозной функцией QoS;
- взаимодействие с существующими сетями посредством открытых интерфейсов;
- универсальная мобильность;
- неограниченный доступ пользователей к разным поставщикам услуг;
- разнообразие схем идентификации;
- единые характеристики обслуживания для одной и той же услуги с точки зрения пользователя;

- сближение услуг между фиксированной и подвижной связью;
- независимость связанных с обслуживанием функций от используемых технологий транспортировки;
- поддержка различных технологий «последней мили»;
- выполнение всех регламентных требований, например, для аварийной связи, защиты информации, конфиденциальности, законного перехвата и т.д.

8.1.2. Преимущества сетей, базирующихся на концепциях NGN

Сети NGN имеют ряд преимуществ (как для пользователей, так и для операторов связи) по сравнению с другими сетями:

- для оператора:
  - построение одной универсальной сети для оказания различных услуг;
  - возможность оптимального использования полосы пропускания для интеграции различных видов трафика и оказания различных услуг;
  - больше возможностей по расширению сети и спектра услуг;
  - простота в управлении и эксплуатации;
  - возможность быстрого внедрения новых услуг и приложений с различным требованием к объёму передаваемой информации и качеству её передачи;
- для пользователя:
  - абстрагирование от технологий реализации услуг электросвязи;
  - гибкое получение необходимого набора, объёма и качества услуг;
  - мобильность получения услуг.

8.1.3. Спектр предоставляемых услуг

В сетях NGN могут предоставляться следующие услуги:

- услуги службы телефонной связи:
  - местное телефонное соединение,
  - междугороднее телефонное соединение,
  - международное телефонное соединение,
  - передача факсимильных сообщений между терминальным оборудованием пользователей,
  - организация модемных соединений между терминальным оборудованием пользователей,
  - переадресация вызова,
  - индикация вызова,
  - удержание вызова;
- услуги служб передачи данных:
  - выделенный канал передачи данных,

  - постоянный и коммутируемый доступа в сеть Интернет,
  - виртуальные частные сети передачи данных;
- услуги телематических служб:
  - электронная почта,
  - голосовая почта,
  - доступ к информационным ресурсам,
  - телефония по IP-протоколу,
  - аудиоконференция и видеоконференция;
- услуги служб подвижной электросвязи;
- услуги поставщиков информации:
  - видео и аудио по запросу,
  - интерактивные новости,
  - электронный супермаркет,
  - дистанционное обучение и др.

8.1.4. Архитектура NGN

С функциональной точки зрения сеть следующего поколения делят на две плоскости — *плоскость услуг (Service Stratum)* и *транспортную плоскость (Transport Stratum)* (рис. 8.1)[1].

Плоскость услуг включает функции, отвечающие за передачу *услугоориентированных данных (Service-Related Data)*, и функции, отвечающие за управление и эксплуатационную поддержку ресурсов услуг и услуг сети, необходимых для предоставления пользователю услуг и приложений.

Транспортная плоскость включает функции, отвечающие за передачу данных, и функции, отвечающие за управление и эксплуатационную поддержку транспортных ресурсов для передачи этих данных между терминальными устройствами.

Взаимодействие приложений и элементов сети NGN осуществляется через *прикладной сетевой интерфейс (Application Network Interface, ANI).*

Сетевой интерфейс пользователя (User Network Interface, UNI) обеспечивает взаимодействие функций конечного пользователя и элементов сети NGN.

Межсетевой интерфейс (Network Network Interface, NNI) обеспечивает взаимодействие сети NGN с другими сетями.

Так как сеть должна обеспечивать передачу разнородного трафика, в том числе чувствительного к задержкам, то немаловажными

[1] ITU-T Recommendation Y.2011: General principles and general reference model for Next Generation Networks. 2004. URL: `http://www.itu.int/rec/T-REC-Y.2011-200410-I/en`; ITU-T Recommendation Y.2012: Functional requirements and architecture of the NGN, release 1. 2006. URL: `http://www.itu.int/rec/T-REC-Y.2012-200609-I/en`.

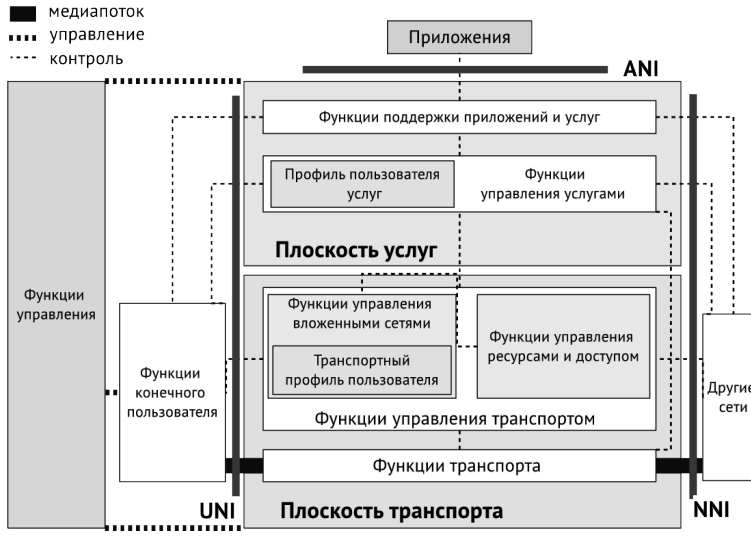


Рис. 8.1. Архитектура NGN

становятся такие требования к сети, как высокая надёжность оборудования узлов, поддержка функций управления трафиком, хорошая масштабируемость.

Функции транспортного уровня

Функции транспортного уровня обеспечивают соединение для всех компонентов и физически разделённых функций в рамках NGN, а также поддержку передачи медиаданных, контрольной и управляющей информации.

Определены следующие функции транспортного уровня:

- функции доступа к сети;
- функции граничного маршрутизатора;
- функции транзитного маршрутизатора;
- функции шлюза;
- функции обработки медианных;
- функции управления транспортным уровнем.

Функции доступа к сети (Access network functions) отвечают за доступ конечных пользователей в сеть, а также за сбор и оценку трафика, полученного транзитным узлом от пользователей. Кроме того, функции доступа к сети осуществляют управление качеством

обслуживания, включая управление ёмкостью буфера, планирование и управление очередью, фильтрацию и классификацию трафика, его маркировку, применение политик по формированию трафика.

Функции доступа к сети классифицируются по технологиям доступа. Соответственно могут быть функции сетевого доступа по кабелю, оптоволокну, xDSL, по беспроводному соединению (IEEE 802.11 и 802.16 технологии, 3G RAN-доступ).

Функции граничного маршрутизатора (Edge functions) используются для обработки данных и трафика в случае, когда трафик, приходящий из различных сетей доступа, сливается в один поток на границе домена NGN. Сюда входят функции, связанные с поддержкой QoS и контролем трафика.

Функции транзитного маршрутизатора (Core transport functions) отвечают за передвижение информации через сеть NGN и предоставляют средства для разделения трафика относительно требований к качеству обслуживания. Данные функции предоставляют QoS-механизмы, непосредственно связанные с трафиком пользователя, включая управление буфером, размером очереди и планированием, фильтрацию пакетов, классификацию трафика, маркировку, разработку политик, контроль за точками доступа и возможностями брандмауэра.

Функции шлюза (Gateway functions) обеспечивают взаимодействие между функциями конечного пользователя и/или другими сетями, включая сети NGN, а также существующие сети, такие как, например, PSTN/ISDN, Интернет и т.д.

Функции обработки медианных (Media handling functions) предоставляют медиаресурсы, необходимые для предоставления услуг, таких как генерация тональных сигналов и преобразование одного кода в другой.

Функции управления транспортным уровнем (Transport control functions) включают в себя *функции контроля доступом к ресурсам (Resource and admission control functions, RACF)* и *функции контроля сетевых подключений (Network attachment control functions, NACFs).*

Функции контроля доступа к ресурсам делают возможным представление для *функций управления услугами (Service Control Functions, SCF)* инфраструктуры транспортной сети в абстрактном виде и освобождают провайдеров от знания таких деталей, как топология сети, интерфейс подключения, потребление ресурсов, механизмы QoS. Данные функции осуществляют контроль за ресурсами сети на основе заданной политики, обеспечивают резервирование ресурсов, взаимодействуют с функциями маршрутизатора с целью контроля за выполнением функций по фильтрации пакетов, классификации трафика, маркировке, определению политики, управлению приоритетами и т.д.

Функции контроля сетевых подключений осуществляют регистрацию пользователя на уровне доступа и инициализацию функций пользователя, необходимых для доступа к услугам NGN. Кроме того, они

идентифицируют транспортный уровень, управляют адресным пространством сети, аутентифицируют сессию доступа.

Таким образом, функции контроля сетевых подключений обеспечивают:
- динамическое предоставление IP-адресов и других параметров конфигурации;
- определение возможностей оборудования пользователя и других параметров;
- аутентификацию пользователя и сети на IP-уровне, а также взаимную аутентификацию пользователя и сети;
- авторизацию доступа в сеть на основе профиля пользователя;
- конфигурацию доступа в сеть на основе профиля пользователя.

Функции уровня управления услугами

Абстрактное представление функциональных групп на уровне управления услугами состоит:
- из функций управления услугами, включая функции профиля пользователя услуги;
- из функций поддержки приложений и функций поддержки услуг.

Функции управления услугами (Service control functions) включают в себя функции управления ресурсами, регистрацией, аутентификацией и авторизацией на уровне услуг. Также могут включать в себя функции управления медиаресурсами, т.е. специализированными ресурсами и шлюзами сигнализации.

Функции управления услугами размещают профили пользователя, представляющие собой информацию о пользователя и другую управляющую информацию, в единый профиль пользователя на уровне услуг в форме базы данных. Эти базы данных могут быть определены и реализованы как набор сообщающихся баз данных с функциональными средствами, расположенными в любой части NGN.

Функции поддержки приложений и функций поддержки услуг (Application support functions and service support functions) включают в себя функции маршрутизации, регистрации, аутентификации, авторизации на уровне приложений. Эти функции доступны как функциональный группе приложений, так и функциональной группе пользователей. Функции поддержки приложений и функции поддержки услуг работают совместно с функциями управления услугами для обеспечения пользователей и приложений теми NGN-услугами, которые им требуются.

Функции конечного пользователя

Интерфейсы пользователей и сетевые интерфейсы, соединённые с сетью доступа NGN, могут быть любыми. Оборудование пользователя может быть как фиксированным, так и мобильным.

Функции управления

Поддержка управления фундаментальна для работы в NGN. Эти функции дают возможность управлять NGN с целью обеспечения NGN услуг ожидаемого качества, безопасности и надёжности.

Функции управления распределены по всем *функциональным модулям (Functional Entity, FE)* и взаимодействуют с сетевыми элементами управления и элементами управления услугами.

Функции управления применяются как на транспортном уровне, так и на уровне услуг NGN. Для каждого уровня они затрагивают следующие области:

- управление исходными настройками;
- управление конфигурациями;
- управление учётными записями пользователей;
- управление производительностью;
- управление безопасностью.

Функции управления учётными записями пользователей дают возможность провайдеру обеспечивать пользователей заказанными ими услугами.

8.1.5. Концепции NGN

Уровень мобильности в архитектуре NGN

Архитектура NGN поддерживает возможность обеспечения мобильности пользователей внутри и между различными сетями доступа и сетями с технологией мобильного доступа. Мобильность может быть поддержана на различных уровнях архитектуры NGN.

Архитектура услуг NGN

Архитектура услуг NGN состоит из трёх различных функциональных областей: области приложений, области функций поддержки приложений и функций поддержки услуг на сервисном уровне, области ресурсов транспортного уровня NGN.

Область функций приложений может быть разбита на две категории — всё, что связано с сетевыми провайдерами, и иное. К первой группе относятся сетевые провайдеры, субпровайдеры и т.д. Ко второй — независимые провайдеры услуг, чей доступ к ресурсам должен быть аутентифицирован, контролируем и профильтрован функциями деблокиратора.

Посредством интерфейса ANI функциональная область функций поддержки приложений и услуг предлагает ресурсы услуг области приложений независимо от технологии сети. Также посредством ANI область приложений получает преимущества от использования возможностей и ресурсов функциональной области инфраструктуры NGN.

Архитектура услуг NGN следует трём основным функциональным характеристикам:

1. агностицизм — области функций поддержки приложений и функций поддержки услуг должны состоять из функций, независимых от инфраструктуры сети NGN;
2. поддержка официальных приложений и черт — архитектура услуг NGN не должна оказывать ограничивающее влияние на саму сеть NGN, т.е. должны поддерживаться функции по управлению сессиями, аутентификация, сведения о местонахождении и т.д.;
3. поддержка открытого интерфейса услуг — платформа услуг NGN должна предоставлять открытый интерфейс услуг (не зависящий от технологий транспортной сети), который обеспечивает доступ к таким функциям, как аутентификация, авторизация и безопасность, чтобы любой провайдер услуг мог воспользоваться возможностями сети.

Функции сокрытия сетевой топологии и просмотра трансляции сетевого адреса и порта

Сокрытие топологии уровня услуг достигается удалением или изменением топологической информации, передаваемой в прикладных сигнальных сообщениях одноранговой сети (например, в SIP-основанных приложениях топологическая информация находится в SIP-заголовках).

Сокрытие топологии транспортного уровня достигается путём изменения топологической информации в пакетах данных или посредством блокировки сетевых контрольных пакетов с топологической информацией (например, изменение IP-адресов и/или номеров портов в пакетах данных, пересекающих границу между сетью доступа и доменом).

Просмотр трансляции сетевого адреса и порта (Network Address and Port Translation, NAPT) осуществляет просмотр удалённого NAPT в сетях доступа.

Контроль за переполнением

Для защиты функциональных модулей управления сессиями от концентрации нежелательных запросов необходима реализация на границе сетей доступа следующих функций: обнаружение концентрации запросов путём сбора информации от двух или нескольких функциональных модулей, передача полученной информации о концентрации запросов другим функциональным модулям, управление трафиком в соответствии с информацией о концентрации запросов.

Функции управления учётными записями пользователей и тарификацией

Функции управления учётными записями пользователей и тарификацией предназначены для представления обобщённой архитектуры предоставления провайдерами услуг пользователям. Они описывают условия сбора и обработки информации о пользователях и заказанных ими услугах для предоставления её NGN-провайдеру. Данные функции включают в себя *функцию сбора данных для тарификации (Charging Trigger Function, CTF)*, *функцию тарификации online (Online Charging Function, OCF)*, *функцию хранения информации для тарификации (Charging Collection Function, CCF)*, *функцию определения стоимости (Rating Function, RF)*, *функцию управления учётными записями пользователей (Account Management Function, AMF)*.

Функция сбора данных для тарификации осуществляет сбор данных о полученных пользователями ресурсах и услугах сети. Кроме того, данная функция создаёт учётные (тарификационные) события, используя онлайновый учёт. Данные направляются онлайновой учётной функции (OCF) для получения авторизации для доступа к ресурсам сети на основе запроса пользователя.

Функция хранения информации для тарификации получает сведения о произошедших событиях от CTF. Затем эта информация используется для формирования тарификационных данных, передаваемых биллинговым доменам.

Функция тарификации online получает данные от CTF и обрабатывают их практически в режиме реального времени для предоставления авторизации использования сетевых ресурсов на основе запроса пользователя. OCF предоставляет квоту на использование ресурсов, которая должна отслеживаться CTF.

Функция определения стоимости определяет стоимость предоставляемого сетевого ресурса.

Функция управления учётной записью хранит баланс учётной записи пользователя во время работы онлайновой учётной функции. Баланс учётной записи пользователя должен определять объём оставшегося доступного трафика, время или содержание, а также количество денег на счёте.

8.1.6. Компоненты сети NGN

На уровне услуг определены два компонента — компонент услуг IP-мультимедиа и компонент эмуляции сервиса PSTN/ISDN.

Компонент услуг IP-мультимедиа предоставляет посреднические услуги, включающие в себя управление услугами реального времени (голосовая или видео телефония, обмен сообщениями и т.п.), основанными на концепции IMS. В NGN IMS расширен для поддержки

дополнительных видов сетей доступа, таких как xDSL и WLAN. Услуга имитации сетей PSTN/ISDN также обеспечивается этим компонентом.

Компонент эмуляции услуг PSTN/ISDN обеспечивает функционирование сетей на основе поддержки существующих услуг для интерфейсов пользователя и оборудования.

Эмуляция PSTN/ISDN относится к предоставлению услуг сетей PSTN/ISDN, используя адаптацию к IP-инфраструктуре. Компонент услуг эмуляции PSTN/ISDN делает возможным поддержку терминалов, связанных с IP-сетью, через шлюз. Все сервисы PSTN/ISDN остаются доступными и идентичными (т.е. с теми же операционными характеристиками), так что пользователи даже не подозревают, что они не соединены с TDM-основанным PSTN/ISDN.

На транспортном уровне также определены два компонента — *компонент функций контроля сетевых подключений (Network Attachment Control Functions, NACF)* и *компонент функций контроля доступом к ресурсам (Resource and Admission Control Functions, RACF)*.

Транспортные сети предоставляют соединения для всех компонентов и физически разделённых функций в рамках NGN. Транспортные сети подразделяются на сети доступа (Access Transport Networks) и внутренние сети (домены) (Core Transport Network) с шлюзом на границе, связывающим транспортные сети этих двух категорий. IP-соединение предоставляется оборудованию пользователя NGN на основе транспортных функций под контролем NACF- и RACF-компонент.

NGN взаимодействует с другими сетями, например, с PSTN/ISDN и Internet. Причём взаимодействие происходит как на уровне услуг, так и на транспортном уровне, посредством граничных шлюзов.

8.1.7. Softswitch и IMS как концепции NGN

Softswitch и IMS реализуют концепцию NGN. Однако следует отметить их принципиально разные подходы к реализации принципов NGN.

Softswitch ориентируется на жёсткую структуру построения сети — регламентируются структура сети, интерфейсы, все компоненты сети (протоколы, кодеки и пр.). При организации сети на базе Softswitch подлежат выполнению все требования стандарта.

IMS демонстрирует модульную архитектуру, регламентируя интерфейсы, оставляя свободу выбора в компонентах (протоколах, кодеков и пр.). Сеть на базе IMS можно реализовывать постепенно, добавляя новые элементы, что позволяет удешевить внедрение.

Подход Softswitch представляется более стройным, в то время как подход IMS — более гибким, что и вызвало ориентацию последних стандартов NGN именно на IMS.

8.2. Сеть на базе стека H.323

Серия рекомендаций H.32x предназначена для организации видеоконференций по различным типам сетей передачи данных (табл. 8.1).

Таблица 8.1

Сводная таблица протоколов семейства H.32x

Рекомендация	**H.320**	**H.321**	**H.322**	**H.323**	**H.324**
Год принятия	1990	1995	1995	1996	1996
Сеть	Узкополосная ISDN	Широкополосная ISDN, ATM	Сеть с коммутацией пакетов и гарантированным QoS (isoEthernet)	Сеть с коммутацией пакетов и негарантированным QoS (Ethernet)	Аналоговые телефонные сети (PSTN или POTS)
Видео	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263 H.264/AVC	H.261 H.263
Аудио	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.723.1 G.726 G.728 G.729	G.723
Мультиплексирование	H.221	H.221	H.221	H.225.0	H.223
Управление	H.230 H.242	H.242	H.242 H.230	H.245	H.245
Поддержка многоточечных конференций	H.231 H.243	H.231 H.243	H.231 H.243	H.323	-
Обмен данными	T.120	T.120	T.120	T.120	T.120
Сетевой интерфейс	I.400	AAL I.363 AJM I.361 PHY I.400	I.400 & TCP/IP	TCP/IP	V.34 модем

H.323 — это рекомендации ITU-T[1] для мультимедийных приложений в вычислительных сетях, не обеспечивающих гарантированное качество обслуживания (QoS). Такие сети включают в себя сети пакетной коммутации IP и IPX на базе Ethernet, Fast Ethernet и пр. Рекомендация H.323 регламентирует технические требования к коммутации речи, видео и данных по пакетным сетям, а также к связи с сетями с коммутацией каналов.

8.2.1. Архитектура сети H.323

Архитектура сети H.323 представлена на рис. 8.2.

Объектами сети H.323 являются:

- *терминал (Terminal)* — оконечное мультимедийное устройство, обеспечивающее возможность двусторонней коммуникации речи, видео или данных с другим объектом сети в реальном времени;
- *межсетевой шлюз (Gateway)* — устройство, предназначенное для преобразования мультимедийной и управляющей информации при сопряжении разнородных сетей;
- *устройство управления многоточечными соединениями (Multipoint Control Unit, MCU)* — предназначено для организации конференций с участием трёх и более участников;
- *контроллер зоны (Gatekeeper)* — рекомендуемое, но не обязательное устройство, обеспечивающее сетевое управление и исполняющее роль виртуальной телефонной станции;
- *разграничитель (Border Element)* — элемент сети H.323, посредством которого выполняется коммуникация между административными доменами.

Терминал H.323 обеспечивает звуковую связь и может дополнительно поддерживать передачу видео или данных. Терминал H.323 может быть реализован как программное приложение на персональном компьютере или как самостоятельное устройство (например, телефон).

Терминал должен поддерживать следующие протоколы:

- H.245 для согласования параметров соединения;
- Q.931 для установления соединения и согласования параметров этого соединения;
- RAS (Registration/Admission/Status) для взаимодействия с контроллером зоны;
- RTP/RTCP для работы с потоками аудио и видеопакетов;
- семейство протоколов H.450;
- аудиокодек G.711 для сжатия аудиопотока.

Дополнительно терминал может поддерживать другие аудиокодеки, а также видеокодеки H.261 и/или H.263. Необязательной является поддержка протокола совместной работы над документами T.120.

[1] ITU-T Recommendation H.323 v1: Packet-based multimedia communications systems. 1996.

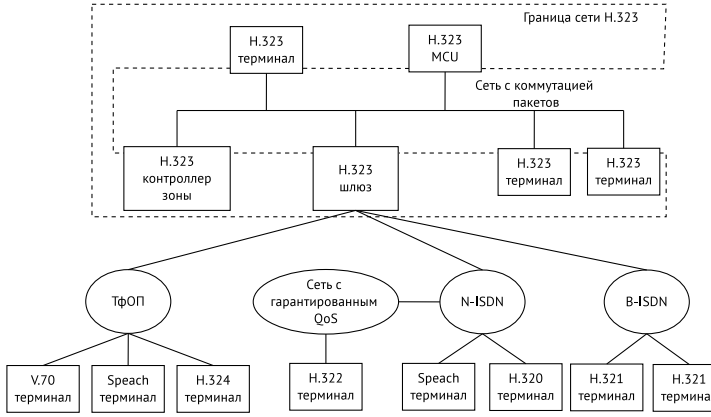


Рис. 8.2. Архитектура сети H.323

Межсетевой шлюз не является обязательным компонентом сети H.323 и используется только в том случае, когда требуется установить соединение с терминалом, расположенным в сети другого типа (стандарта). Связь обеспечивается трансляцией протоколов установки и разрыва соединений, а также форматов передачи данных. Основной функцией межсетевого шлюза H.323 является преобразование сигнализационных протоколов, способа передачи, процедур коммуникации и способа кодирования, что обеспечивает возможность взаимодействия пользователей разных технологий. Шлюзы H.323 широко применяются в IP-телефонии для сопряжения IP-сетей и цифровых или аналоговых коммутируемых телефонных сетей (ISDN (Integrated Services Digital Network) или PSTN (Public Switched Telephone Network)).

Межсетевой шлюз может выполнять следующие функции:

- функция PSTN-терминала — содержит PSTN сигнализационный интерфейс, которым заканчивается PSTN сигнализация, и PSTN медиаинтерфейс, которым заканчивается медиапоток;
- функция H.323-терминала — содержит VoIP сигнализационный интерфейс, которым заканчивается H.323 сигнализация (H.225, H.245), и интерфейс пакетной передачи, которым заканчивается медиапоток, передаваемый пакетами протокола RTP;
- преобразование сигнализационных протоколов, используемых в H.323 и PSTN-сетях;
- преобразование медиапотоков, сформированных при помощи различных алгоритмов сжатия;
- управление связью — координирование сигнализационных потоков и преобразование медиапотоков, в том числе и установление, изме-

нение и разъединение соединения между медиапотоками в PSTN и IP-сети в течение вызова.

Контроллер зоны также не является обязательным компонентом сети H.323, и если используется, то обеспечивает сетевое управление и выполняет функции виртуальной телефонной станции. В этом случае контроллер зоны становится центральной точкой для всех обращений внутри одной зоны — совокупности терминалов, шлюзов и серверов MCU, управляемых одним контроллером.

Контроллер зоны выполняет следующие функции:

- основные:
  - трансляция адресов — преобразование внутренних адресов сети и телефонных номеров формата E.164 (применяются в сетях ISDN) в транспортные адреса протоколов IP или IPX;
  - управление доступом — авторизация доступа в H.323-сеть путём обмена RAS-сообщениями «запрос регистрации» (ARQ), «удовлетворение запроса» (ACF) и «отклонение запроса» (ARJ);
  - управление полосой пропускания — используются RAS-сообщения «запрос ширины полосы пропускания» (BRQ), «удовлетворение запроса» (BCF) и «отклонение запроса» (BRJ);
  - управление зоной H.323 — установление вызова, использование ресурсов разрешается исключительно тем объектам сети H.323, которые зарегистрированы как члены зоны определённого контроллера зоны;
- дополнительные:
  - управление процессом установления соединений — обработка служебных сообщений протокола сигнализации Q.931[1];
  - авторизация соединения;
  - управление вызовами — контроль за состоянием всех активных соединений, что позволяет обеспечить выделение необходимой полосы пропускания и баланс загрузки сетевых ресурсов за счёт переадресации вызовов на другие терминалы и шлюзы;
  - тарификация — хранение и обработка информации о вызовах и предоставленных услугах.

Устройство MCU предназначено для поддержки конференции между тремя и более участниками. В этом устройстве должен присутствовать *контроллер Multipoint Controller (MC)* и, возможно, *процессоры Multipoint Processors (MP)*. Контроллер MC поддерживает протокол H.245[2] и предназначен для согласования параметров обработки аудио- и видеопотоков между терминалами. Процессоры занимаются коммутированием, микшированием и обработкой этих потоков.

---

[1] ITU-T Recommendation Q.931: ISDN user-network interface layer 3 specification for basic call control. 1998.

[2] ITU-T Recommendation H.245, Control protocol for multimedia communication. 2006.

Стандарт H.323 определяет три типа конференцсвязи между тремя или более числом терминалов и межсетевых шлюзов: *централизованная*, *децентрализованная*, *гибридная*.

Централизованная многоточечная конференция требует наличия устройства MCU. Каждый терминал обменивается с MCU потоками аудио, видео, данными и командами управления по схеме «точка–точка». Контроллер MC, используя протокол H.245, определяет возможности каждого терминала. Процессор MP формирует необходимые для каждого терминала мультимедийные потоки и рассылает их. Кроме того, процессор может обеспечивать преобразования потоков от различных кодеков с различными скоростями данных.

Децентрализованная многоточечная конференция использует технологию
групповой адресации. Участвующие в конференции H.323 терминалы осуществляют многоадресную передачу мультимедиа потока остальным участникам без посылки на MCU. Передача контрольной и управляющей информации осуществляется по схеме «точка–точка» между терминалами и MCU. В этом случае контроль многоточечной рассылки осуществляется контроллером MC.

Гибридная схема организации конференцсвязи является комбинацией двух
предыдущих. Участвующие в конференции H.323 терминалы осуществляют многоадресную передачу только аудио- или только видеопотока остальным участникам без посылки на MCU. Передача остальных потоков осуществляется по схеме «точка–точка» между терминалами и MCU. В этом случае задействуются как контроллер, так и процессор MCU.

8.2.2. Адресация элементов сети H.323

Адресация терминалов VoIP в основном основывается на буквенно-цифровых последовательностях, распознаваемость которых обеспечена иерархической организацией группы серверов. Однако из-за потребности интеграции услуг между сетями PSTN и VoIP каждому абоненту PSTN должна быть обеспечена возможность адресации VoIP абонента, и наоборот.

Стандарт H.323 поддерживает следующие типы адресов:

- dialedDigits (в старых версиях E.164) — цифровой идентификатор в виде телефонного номера;
- h323-ID — имя пользователя или адрес электронной почты (e-mail address);
- url-ID — общий тип адреса (включает H.323-URL и PSTN-URL);
- transport-ID — транспортный адрес оконечного оборудования;
- email-ID — адрес электронной почты;
- partyNumber — цифровой идентификатор;

– mobile-UIM — идентификатор мобильных пользователей с возможностью взаимодействия с мобильными сетями общего пользования 2G и 3G.

Такой подход требует отдельного преобразования и распознавания адреса, а также особых процедур регистрации, обеспечиваемых контроллерами зоны H.323 и разграничителями.

В случае адресации с помощью цифр телефонного номера используются префикс зоны и технологический префикс, однозначно определяющие зону административного домена.

Каждое устройство в сети H.323 может иметь более одного адреса (возможно, одного и того же типа). Единственное условие — все адреса одного устройства должны ссылаться на уникальный транспортный адрес этого устройства.

8.2.3. Основные характеристики H.323

Основные характеристики H.323:
– независимость от сети — возможна работа поверх существующих архитектур сетей;
– управление шириной полосы — каждому H.323-вызову выделяется определённая ширина полосы;
– независимость от приложения и платформы — не требуется применения определённой аппаратной или программной платформы;
– поддержка многосторонних конференций;
– взаимодействие — прозрачная коммутация для конечного пользователя;
– гибкость — в одной H.323 конференции могут участвовать терминалы различных возможностей коммуникации.

8.2.4. Обработка звуковых сигналов (Audio Signal)

Одним из важных факторов эффективного использования пропускной способности IP-канала является выбор оптимального алгоритма кодирования / декодирования речевой информации — кодека.

Типы речевых кодеков по принципу действия можно разделить на три группы:
1. кодеки с импульсно-кодовой модуляцией (ИКМ) и адаптивной дифференциальной импульсно-кодовой модуляцией (АДИКМ):
   – разработаны в конце 1950-х годов,
   – используются в системах традиционной телефонии;
2. кодеки с вокодерным[1] преобразованием речевого сигнала:

[1]Вокодер — электронный цифровой музыкальный инструмент, преобразующий звук человеческого голоса путём изменения его волновых и частотных характеристик.

- разработаны для снижения требований к пропускной способности радиотракта в системах мобильной связи,
- применяется гармонический синтез сигнала на основании информации о его вокальных составляющих — фонемах,
- реализованы как аналоговые устройства;

3. комбинированные (гибридные) кодеки:
   - сочетают в себе технологию вокодерного преобразования / синтеза речи, но оперируют с цифровым сигналом посредством специализированных преобразователей цифровых сигналов,
   - содержат в себе ИКМ или АДИКМ кодек и реализованный цифровым способом вокодер.

Оценка MOS

MOS (Mean Opinion Score) — средняя экспертная оценка разборчивости речи —- метод субъективного тестирования качества речи, часто используемый для сравнения характеристик речевых кодеков, при котором слушатели выставляют оценки по пятибалльной системе. Результирующая оценка MOS вычисляется как среднее арифметическое для большого числа оценок.

Таблица 8.2

Оценки MOS

Качество	Оценка MOS
высокое	4, 0–5, 0
стандартное телефонное	3, 5–4, 0
приемлемое	3, 0–3, 5
синтезированный звук	2, 5–3, 0

G.711

Рекомендация G.711[1] описывает кодек, использующий преобразование аналогового сигнала с точностью 8 бит, тактовой частотой 8 кГц и простейшей компрессией амплитуды сигнала. Скорость потока данных на выходе преобразователя составляет 64 Кбит/с (8 бит х 8 кГц). Для снижения шума квантования и улучшения преобразования сигналов с небольшой амплитудой при кодировании используется нелинейное

[1] ITU-T Recommendation G.711 Pulse code modulation (PCM) of voice frequencies. 1988.

квантование по уровню согласно специальному псевдологарифмическому закону. Существуют два основных алгоритма, представленных в стандарте:

1. μ-law (используется в Северной Америке и Японии):
   прямое преобразование:

$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 \leqslant x \leqslant 1;$$

   обратное преобразование:

$$F^{-1}(y) = \operatorname{sgn}(y)\frac{1}{\mu}\left[(1+\mu)^{|y|} - 1\right], \quad -1 \leqslant y \leqslant 1,$$

   где $\mu = 255$ (8 бит).
2. A-law (используется в Европе и в остальном мире):
   прямое преобразование:

$$F(x) = \operatorname{sgn}(x)\begin{cases} A\dfrac{|x|}{1+\ln(A)}, & |x| \leqslant \dfrac{1}{A}, \\ \dfrac{1+\ln(A|x|)}{1+\ln(A)}, & \dfrac{1}{A} \leqslant |x| \leqslant 1; \end{cases}$$

   обратное преобразование:

$$F^{-1}(y) = \operatorname{sgn}(y)\begin{cases} \dfrac{|y|(1+\ln(A))}{A}, & |y| \leqslant \dfrac{1}{1+\ln(A)}, \\ \dfrac{\exp(|y|(1+\ln(A))-1)}{A}, & \dfrac{1}{1+\ln(A)} \leqslant |y| < 1, \end{cases}$$

   где $A = 87,6$ — параметр сжатия.

Оба алгоритма являются логарифмическими, но более поздний A-law был изначально предназначен для компьютерной обработки процессов. Типичная оценка MOS составляет $4,2$. Обычно любое устройство VoIP поддерживает этот тип кодирования.

Кодек G.711 широко распространён в системах традиционной телефонии с коммутацией каналов. Несмотря на то, что рекомендация G.711 в стандарте H.323 является основной и первичной, в шлюзах IP-телефонии данный кодек применяется редко из-за высоких требований к полосе пропускания и задержкам в канале передачи. Использование G.711 в системах IP-телефонии обоснованно лишь в тех случаях, когда требуется обеспечить максимальное качество кодирования речевой информации при небольшом числе одновременных разговоров.

G.723.1

Кодек G.723.1[1] является одним из базовых кодеков сжатия речи, утверждённым ITU-T в рекомендации G.723.1 в ноябре 1995 г. Кодек предназначен для приложений IP-телефонии, в частности, для организации видеоконференций по телефонным сетям. Рекомендация G.723.1 является частью более общего стандарта H.324, описывающего подход к организации видеоконференций, при этом целью является обеспечение видеоконференций с использованием обычных модемов.

Кодек G.723.1 представляет собой комбинацию аналого-цифрового преобразования / цифро-аналогового преобразования и вокодера. Применение вокодера позволяет снизить скорость передачи данных в канале, что принципиально важно для эффективного использования как радиотракта, так и IP-канала.

Кодек G.723.1 осуществляет преобразование аналогового сигнала в поток данных со скоростью 64 Кбит/с (ИКМ), а затем при помощи многополосного цифрового фильтра / вокодера выделяет частотные фонемы, анализирует их и передаёт по IP-каналу информацию только о текущем состоянии фонем в речевом сигнале. Данный алгоритм преобразования позволяет снизить скорость кодированной информации до 5,3–6,3 Кбит/с без видимого ухудшения качества речи.

Кодек G.723.1 предусматривает два режима работы: 6,3 Кбит/с (кадр имеет размер 189 бит, дополненных до 24 байт) и 5,3 Кбит/с (кадр имеет размер 158 бит, дополненных до 20 байт). Первый режим применяется для сетей с пакетной передачей голоса и использует алгоритм сжатия речи MP-MLQ (Multipulse Maximum Likelihood Quantization — - многоимпульсное квантование с максимальным правдоподобием), который позволяет добиться весьма существенного сжатия речевой информации при сохранении достаточно высокого качества звучания. Второй режим применяется в сетях со смешанным типом трафика (голос / данные) и использует алгоритм CELP (Code Excited Linear Prediction — кодирование с линейным предсказанием). Режим работы кодека G.723.1 может меняться динамически от кадра к кадру.

Алгоритм CELP[2] построен на модели кодирования с использованием процедуры «анализа через синтез», линейного предсказания и векторного квантования. CELP-анализ состоит из трёх основных процедур:

---

[1] ITU-T Recommendation G.723.1 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. 1996, 2006.

[2] Federal Standard 1016, Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP). Washington : National Communications System, Office of Technology, Standards, 1991; NCS Technical Information Bulletin 92-1. Details to Assist in Implementation of Federal Standard 1016 CELP.

– кратковременное линейное предсказание;
– долговременный поиск по адаптивной кодовой книге;
– поиск по стохастической кодовой книге.

CELP-синтез состоит из этих же процедур, выполненных в обратном порядке.

Кодер оперирует с кадрами речевого сигнала длиной 30 мс, дискретизованными с частотой 8 кГц. Для каждого кадра производится анализ речевого сигнала и выделяются передаваемые параметры CELP-модели: 10 линейных спектральных пар (несут информацию о коэффициентах фильтра линейного предсказания), индексы и коэффициенты усиления в адаптивной и фиксированной кодовых книгах. Далее эти параметры кодируются в битовый поток и передаются в канал.

В декодере эта битовая посылка используется для восстановления параметров сигнала возбуждения и коэффициентов синтезирующего фильтра. Далее восстанавливается речь путём пропускания сигнала возбуждения через синтезирующий фильтр. Затем для улучшения качества восприятия синтетического сигнала выходной сигнал с фильтра-синтезатора пропускается через постфильтр.

Длительность кадров кодека G.723.1 составляет 30 мс с длительностью предварительного анализа сигнала $7,5$ мс.

Оценка MOS для данного кодека составляет $3,9$ в режиме $6,3$ Кбит/с и $3,7$ — в режиме $5,3$ Кбит/с.

Кодек G.726

Рекомендация G.726 основана на алгоритме кодирования ADPCM — адаптивная дифференциальная ИКМ. Этот алгоритм даёт практически такое же качество воспроизведения речи, как и ИКМ, однако для передачи информации при его использовании требуется полоса всего 16–32 Кбит/с. Кодек предназначен для использования в системах видеоконференций; в приложениях IP-телефонии этот кодек практически не применяется. Оценка по MOS составляет $4,3$.

G.728

Алгоритм G.728 стандартизован ITU в 1992 г.[1], основан на методе LD-CELP (Low-Delay Code Excited Linear Prediction — кодирование с линейным предсказанием и низкой задержкой) и предназначен для сжатия и передачи речевых данных со скоростью 16 Кбит/с, при этом внося задержку при кодировании от 3 до 5 мс.

[1] ITU-T Recomendation G.728 Coding of speech at 16 kbit/s using law delay code excited linear prediction. 1992.

Алгоритм применяется к цифровой последовательности, получаемой в результате аналого-цифрового преобразования речевого сигнала с 16-разрядным разрешением. Входной сигнал с частотой дискретизации 8кГц, сжатый по A- или μ-закону (см. раздел 8.2.4), преобразуется для получения линейного кода.

Оценка MOS для данного кодека составляет $3,6$.

Предназначен для использования в основном в системах видеоконференций. В устройствах IP-телефонии данный кодек применяется достаточно редко.

G.729

В основе кодеков G.729[1] лежит алгоритм CS-ACELP (Conjugate Structure – Algebraic Code Excited Linear Prediction) — сопряжённая структура с управляемым алгебраическим кодированием с линейным предсказанием. Процесс преобразования вносит задержку 15 мс. Скорость кодированного речевого сигнала составляет 8 Кбит/с.

Алгоритм основан на модели кодирования с использованием линейного предсказания по алгебраической кодовой книге (CELP-модель). Кодер оперирует с кадрами речевого сигнала длительностью 10 мс, дискретизованными с частотой 8кГц, что соответствует 80 16-битным отсчётам в линейном законе. Для каждого кадра производится анализ речевого сигнала и выделяются параметры модели (коэффициенты фильтра линейного предсказания, индексы и коэффициенты усиления в адаптивной и фиксированной кодовых книгах). Далее эти параметры кодируются и передаются в канал.

В декодере битовая посылка используется для восстановления параметров сигнала возбуждения и коэффициентов синтезирующего фильтра. Речь восстанавливается путём пропускания сигнала через кратковременный синтезирующий фильтр.

Синтезирующий фильтр имеет полюсную передаточную функцию 10-го порядка. Для работы синтезатора основного тона используется адаптивная кодовая книга. В последующем речь улучшается адаптивной постфильтрацией.

В случае потери передаваемой кодером битовой посылки исходные данные для речевого синтезатора получаются интерполяцией данных

[1]ITU-T Recommendation G.729 Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction. 1996; ITU-T Recommendation G.729 — Annex A, Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACLEP), Annex A: Reduced complexity 8 kbit/s CS-ACELP speech codec. 1996; ITU-T Recommendation G.729 — Annex B. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACLEP), Annex B: A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70. 1996.

с предыдущих не повреждённых кадров, но при этом энергия интерполированного речевого сигнала постепенно уменьшается, что не создаёт особого дискомфорта у слушателя.

В устройствах VoIP и VoFR данный кодек занимает лидирующее положение, обеспечивая наилучшее качество кодирования речевой информации при достаточно высокой компрессии.

8.2.5. Обработка видеосигналов (Video Signal)

Стандарт H.323[1] устанавливает два формата изображения —- CIF (352×288 пиксела) для яркостного сигнала и QCIF (176×144 пиксела), т.е. с 1/4 частью разрешения CIF, причём частота смены кадров не должна опускаться ниже 24 кадров в секунду.

CIF (Common Itermediate Format — общий формат обмена) представляет собой стандарт видеоизображения с размером кадра 352×288 пиксела и частотой кадров $7, 5$, 10, 15 или 30 к/с. Цвет кодируется в формате YUV (представление цвета, при котором каждый элемент изображения представляется тремя компонентами: яркостной и двумя цветоразностными) с разрядностью 8 бит. Производные форматы: QCIF — 176×144 пикселов, subQCIF — 128×96 пикселов, 4CIF — 704×576 пикселов, 16CIF — 1408×1152 пикселов.

Для компрессии/декомпрессии видеосигнала используются кодеки H.261, H.263, H.264. Различаются они способом обработки изображения.

H.261

Стандарт H.261[2] определяет видеокодек H.261 для аудиовизуальных услуг со скоростью $P \times 64$ Кбит/с, где P может меняться в диапазоне от 1 до 30. В данном кодеке реализована комбинация алгоритмов *DCT (Discrete Cosine Transform)* и *Motion Prediction*.

Алгоритм *DCT (Discrete Cosine Transform — дискретное косинус-преобразование, ДКП)* разработан в 1981 г. В[3] даётся следующее определение.

---

[1] ITU-T Recommendation H.323 v1: Packet-based multimedia communications systems. 1996.

[2] ITU-T Recommendation H.261: Video CODEC for audiovisual services at p X 64 kbit/s. 1993.

[3] Discrete-time signal processing. 2nd edition Upper Saddle River. NJ, USA : Prentice-Hall, Inc., 1999; Алгоритмические основы растровой графики / Д. В. Иванов, А. С. Карпов, Е. П. Кузьмин, В. С. Лемпицкий, А. А. Хропов. Изд. ИНТУИТ, 2007.

Определение. Пусть дано изображение размером $N \times N$. Тогда прямое ДКП записывается в виде:

$$t(u,v) = c(u)c(v)\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} I(k,l)\cos\frac{(2k+1)u\pi}{2N}\cos\frac{(2l+1)v\pi}{2N},$$

$$c(m) = \begin{cases} \sqrt{\frac{1}{N}}, m = 0, \\ \sqrt{\frac{2}{N}}, m \neq 0, \end{cases} \quad u,v = \overline{1,N-1},$$

а обратное:

$$I(k,l) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} t(u,v)c(u)c(v)\cos\frac{(2k+1)u\pi}{2N}\cos\frac{(2l+1)v\pi}{2N},$$

$$c(m) = \begin{cases} \sqrt{\frac{1}{N}}, & m = 0, \\ \sqrt{\frac{2}{N}}, & m \neq 0. \end{cases} \quad k,l = \overline{1,N-1},$$

Здесь коэффициенты $t(u,v)$ — амплитуды пространственных частот изображения.

Дискретное преобразование обладает следующими свойствами:

- некоррелированность коэффициентов — коэффициенты независимы друг от друга, т.е. точность представления одного коэффициента не зависит от любого другого;
- «уплотнение» энергии — преобразование сохраняет основную информацию в малом количестве коэффициентов.

Motion Prediction —- *предсказание перемещения* — техника межкадрового кодирования, применяемая в кодеках для сжатия сигнала движущегося изображения. В последовательности кадров каждый пиксель в текущем кадре перемещён по сравнению с предшествующим кадром. При этом соседние пиксели перемещаются практически одинаково. Кадр делится на блоки пикселей (16×16 или 8×8), и для описания движения пикселей всего блока вычисляется вектор оценки перемещения (*Motion Estimation*). Предсказание перемещения текущего блока, полученное из предшествующего кадра с помощью вектора компенсации перемещения (*Motion Compensation*), сравнивается с настоящим текущим блоком и формируется, если надо, ошибка предсказания (т.е. компенсация неточности предсказания). Для таких блоков передаётся только вектор оценки перемещения и ошибка предсказания, что значительно экономней простой передачи содержимого блока.

H.263

Стандарт H.263[1] разработан в 1995 г. и определяет видеокодек H.263, предназначенный для передачи видеоизображения с малой скоростью (ниже 64 Кбит/с, например, для связи с помощью модема и аналоговых телефонных линий). Кодек H.263 использует технологию H.261 с дополнительными усовершенствованиями, главным образом в области предсказания перемещения. В отличие от H.261, для которого предсказываемые направления должны лежать в пределах изображения, для H.263 они могут выходить за границы изображения. Это особенно важно при низких скоростях передачи, не являющихся обязательными для стандарта H.261. Кроме того, кодек H.263 позволяет загружать канал связи практически только изменениями картинки.

Дальнейшим развитием проекта являются кодеки H.263v2 (также известный как H.263+ или H.263 1998) и H.263v3 (известный как H.263++ или H.263 2000).

H.264

Стандарт H.264[2] разработан совместно ITU-T и MPEG и является развитием H.263. Он определяет одноимённый кодек H.264, также известный как *AVC (Advanced Video Coding)* и *MPEG-4*[3], который имеет существенно расширенные возможности по сравнению с H.263, вследствие чего стал основным при разработке программного обеспечения для видеоконференций.

Основные характеристики H.264:

- Многокадровое предсказание перемещения кадров:
  - Более гибкое использование сжатых ранее кадров в качестве опорных. Разрешается использование до 32 ссылок на другие кадры, что поднимает эффективность кодирования, так как позволяет кодеру выбирать для компенсации движения между большим количеством изображений.
  - Независимость порядка воспроизведения изображений и порядка опорных изображений, что позволяет кодеру выбирать порядок изображений для компенсации движения и для воспроизведения с высокой степенью гибкости, которая ограничена только объёмом памяти, гарантирующим возможность декодирования. Устранение ограничения также позволяет в ряде случаев устранить

[1] ITU-T Recommendation H.263: Video coding for low bit rate communication. 2005.

[2] ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services. 2005.

[3] ISO/IEC 14496-10 Standart «Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding». 2005.

дополнительную задержку, ранее связанную с двунаправленным предсказанием.
  - Независимость методов обработки изображений и возможности их использования для предсказания движения, что обеспечивает кодеру большую гибкость и возможность использовать для предсказания движения изображение, более близкое по содержанию к кодируемому.
  - Компенсация движения с переменным размером блока (от 16×16 до 4×4 пикселя) позволяет крайне точно выделять области движения.
  - Вектора движения, выводящие за границы изображения (по аналогии с H.263).
  - Шеститочечная фильтрация компонента яркости для полупиксельного предсказания с целью уменьшения зубчатости краёв и обеспечения большей чёткости изображения.
  - Точность до четверти пиксела при компенсации движения обеспечивает очень высокую точность описания движущихся областей (что особенно актуально для медленного движения).
  - Взвешенное предсказание, позволяющее использовать масштабирование и сдвиг после компенсации движения на величины, указанные кодером. Такая методика может чрезвычайно сильно поднять эффективность кодирования для сцен с изменением освещённости, например, при эффектах затемнения, постепенного появления изображения.
- Пространственное предсказание от краёв соседних блоков для I-кадров (от англ. Intra Pictures). Новая методика экстраполяции краёв ранее декодированных частей текущего изображения повышает качество сигнала, используемого для предсказания.
- Сжатие макроблоков без потерь:
  - Метод представления макроблоков без потерь в ИКМ, при котором видеоданные представлены непосредственно, что позволяет точно описывать определённые области и допускать строгое ограничение на количество закодированных данных для каждого макроблока.
  - Улучшенный метод представления макроблоков без потерь, позволяющий точно описывать определённые области, затрачивая при этом существенно меньше битов, чем ИКМ.
- Гибкие функции чересстрочного сжатия:
  - Адаптивное к изображению кодирование полей (PAFF), позволяющее кодировать каждый кадр как кадр или как пару полей (полукадров) — в зависимости от отсутствия/наличия движения.
  - Адаптивное к макроблокам кодирование полей (MBAFF), позволяющее независимо кодировать каждую вертикальную пару макроблоков (блок 16×32) как прогрессивные или чересстрочные. Позволяет использовать макроблоки 16×16 в режиме разбиения

на поля.
- Новые функции преобразования:
  - Точное целочисленное преобразование пространственных блоков 4×4, позволяющее точно разместить разностные сигналы с минимумом шума.
  - Точное целочисленное преобразование пространственных блоков 8×8, обеспечивающее большую эффективность сжатия схожих областей, чем 4×4.
  - Адаптивный выбор кодеком между размерами блока 4×4 и 8×8.
- Дополнительное преобразование Адамара (разложение обрабатываемых сигналов по системе прямоугольных базисных функций), применяемое к дискретно-косинусным коэффициентам основного пространственного преобразования (к коэффициентам яркости и, в особом случае, цветности) для достижения большей степени сжатия в однородных областях.
- Квантование:
  - Логарифмическое управление длиной шага для упрощения распределения битрейта (битовая скорость передачи данных) кодером и упрощённого вычисления обратной функции квантования.
  - Частотно-оптимизированные матрицы масштабирования квантования, выбираемые кодером для оптимизации квантования на основе человеческих особенностей восприятия.
- Внутренний фильтр деблокинга (удаление блочности) в цикле кодирования, устраняющий артефакты (искажение) блочности, часто возникающие при использовании основанных на DCT-техниках сжатия изображений.
- Энтропийное кодирование[1] квантованных коэффициентов трансформации:
  - *Context-adaptive binary arithmetic coding (CABAC) —- контекстнозависимое адаптивное бинарное арифметическое кодирование* — алгоритм сжатия без потерь синтаксических элементов видеопотока на основе вероятности их появления.
  - *Context-adaptive variable-length coding (CAVLC) —- контекстнозависимое адаптивное кодирование с переменной длиной кодового слова* — альтернатива CABAC меньшей сложности.
  - Часто используемое, простое и высокоструктурированное кодирование словами переменной длины многих элементов синтаксиса, не закодированных CABAC или CAVLC, известное как Exp-Golomb (экспоненциальное кодирование Голомба).
- Функции устойчивости к ошибкам:

[1] *Энтропийное кодирование* — кодирование словами (кодами) переменной длины, при которой длина кода символа имеет обратную зависимость от вероятности появления символа в передаваемом сообщении.

  - Определение уровня сетевой абстракции, позволяющее использовать один и тот же синтаксис видео в различных сетевых окружениях, включая наборы параметров последовательности и наборы параметров изображения, которые обеспечивают большую надёжность и гибкость, чем предыдущие технологии.
  - Гибкое упорядочивание макроблоков, также известное как группы частей и произвольное упорядочивание частей — методы реструктурирования порядка представления макроблоков в изображениях.
- Благодаря произвольному упорядочиванию частей новый стандарт позволяет посылать и получать их в произвольном порядке друг относительно друга. Это может снизить задержку в приложениях реального времени.
  - Разбиение данных — функция, обеспечивающая разделение данных разной важности по разным пакетам данных с разными уровнями защиты от ошибок.
  - Избыточные части. Возможность посылки кодером избыточного представления областей изображений, позволяя воспроизвести области изображений, данные о которых были потеряны в процессе передачи.
  - Нумерация кадров, позволяющая создать «подпоследовательности»
    (включая временное масштабирование включением дополнительных кадров между другими), а также обнаружить (и скрыть) потери целых кадров при сбоях канала или пропаже пакетов.

8.2.6. Конференц-связь для передачи данных (Data)

Стандарт T.120[1] представляет собой совокупность телекоммуникационных и прикладных протоколов для организации и проведения многоточечной конференции в реальном времени [10].

Данный стандарт регламентирует порядок организации и поддержания конференций на любой платформе, управление множеством участников и программ, безошибочный и безопасный обмен данными при различных возможных сетевых сценариях.

В семейство T.120 входят следующие протоколы:

- T.121 представляет основу для разработки прикладных протоколов;
- T.122 совместно с T.125 определяет доступные многоточечные услуги;
- T.123 специфицирует транспортные профили ТфОП, ISDN, цифровых сетей с коммутацией каналов CSDN, цифровых сетей с коммутацией пакетов PSDN, сети Novell NetWare IPX и сети TCP/IP; обеспечивает вышележащим уровням независимость от типа сети

[1] ITU-T Recommendation T.120: Transmission protocols for multimedia data. 1996.

и предоставляет четыре канала разного приоритета между двумя точками, что необходимо для обеспечения преимущества пересылки данных реального времени (например, информации о перемещении курсора) перед фоновой передачей данных (например, транспортировкой файлов);
- T.124 регламентирует общий процесс управления конференцией Generic Conference Control (GCC), обеспечивая полный набор инструментов для её организации и управления; в частности, GCC обеспечивает функции ведущего конференции и функции резервирования.
- T.125 описывает многоточечный протокол связи (Multipoint Communication Service Protocol, MCS), задающий процедуры для передачи сигнальной информации и данных между провайдерами MCS; при многоточечном соединении можно ограничить доступ к определённым наборам данных, сделав их доступными лишь для некоторых участников телеконференции;
- T.126 определяет процедуры просмотра и аннотирования неподвижных изображений между двумя или несколькими приложениями;
- T.127 предусматривает средства файлового обмена между участниками конференции, в том числе их одновременную приоритетную передачу, а также опции для сжатия файлов перед их транспортированием;
- T.128 регламентирует аудиовизуальное управление.

Стек протоколов T.120 имеет двухуровневую архитектуру. Протоколы T.122, T.123, T.124 и T.125 образуют нижний уровень и описывают независимый от приложений механизм для организации многоточечной связи. В тоже время, протоколы T.126, T.127 и T.128 располагаются на верхнем уровне и по своей сути являются прикладными протоколами. Следует отметить, что в рамках одной конференции могут сосуществовать как стандартизованные, так и нестандартизованные приложения.

В зависимости от конкретной реализации продукты T.120 могут устанавливать соединения, выполнять передачу и приём данных и работать совместно, используя программное разделение, передачу файлов и др.

8.2.7. Управление (Control)

Совокупная система управления H.323 основывается на трёх отдельных сигнализационных каналах: канале H.245, канале установления вызова и RAS-канале.

Протокол управления мультимедийной конференцией H.245[1] обеспечивает согласование возможностей компонентов, установление и

[1]ITU-T Recommendation H.245, Control protocol for multimedia communication. 2006.

разрыв логических соединений, передачу запросов на установление приоритета, управление потоком (загрузкой канала), передачу общих команд и индикаторов.

Сообщения протокола H.245 передаются по H.245-каналу управления, используя коммутируемый способ передачи данных с помощью протокола TCP, что гарантирует последовательную передачу данных без ошибок. Между любыми двумя элементами сети можно установить только один H.245-канал.

Межтерминальный обмен параметрами позволяет согласовывать режимы работы и форматы кодирования информации, что обеспечивает взаимодействие терминалов от разных производителей. В процессе обмена сообщениями о параметрах уточняются возможности терминалов по приёму и передаче различных видов трафика.

Все H.245-сигнализационные сообщения принадлежат одной из следующих категорий:

- запрос (Request) — сообщения, которые требуют от получателя выполнения определённых действий, включая и ответ на принятый запрос;
- ответ (Response) — сообщения, которые посылаются в ответ на сообщения из предыдущей категории;
- команда (Command) — команды, которые от получателя требуют выполнения определённых действий, но не включают ответ на команду;
- индикация (Indication) — сообщения информативного типа, которые от получателя не требуют ни действий, ни ответа.

Процедуры H.245:

- объявление о возможности обмена медиа потоками (Capabilities Exchange) — информация, необходимая для выбора поддерживаемого обеими сторонами вида медиа коммуникации;
- определение ведущей стороны в коммуникации (Master Slave Determination) — договорённость о ведущем и ведомых оконечных узлах;
- открытие и закрытие логических каналов сигнализации (Logical Channel Signalling);
- запрос на изменение установленного соединения (Request Mode) — запрос на изменение характеристик медиа потока;
- закрытие канала H.245.

Если канал установления вызова ненадёжный, то для обмена сигнализацией применяется протокол H.225.0[1], используя некоммутируемый способ передачи данных с помощью протокола UDP. В этом случае для установления вызова определён отдельный механизм подтвержде-

[1]ITU-T Recommendation H.225.0, Call signalling protocols and media stream packetization for packet-based multimedia communication systems. 2006.

ния приёма и повторной передачи, т.к. для сигнализации, связанной с установлением вызова, требуется надёжная передача.

Протокол H.225.0 представляет собой протокол сигнализации для установления и разъединения H.323 вызова между двумя H.323 оконечными точками. В рамках этого протокола определена и процедура ускоренного соединения (Fast Connect Procedure).

Протокол H.225.0 в рамках процедур, требуемых для установления и разъединения вызова, определяет использование следующих сообщений:

- Setup — сообщение о начале установления соединения;
- Setup Acknowledge — подтверждение установления соединения;
- Information — информация, необходимая для установления вызова, или другие сведения, относящиеся к вызову;
- Call Proceeding — сообщение о продолжении установления вызова;
- Progress — в этом сообщении посылается информация о дальнейшем развитии вызова при взаимодействии с сетями с коммутацией каналов;
- Alerting — оповещение о входящем вызове;
- Connect — сообщение об установлении соединения;
- Facility — сообщение для осуществления дополнительных услуг и туннелирования H.245-сообщений по каналам установления вызова;
- Status Inquiry — запрос статуса вызова;
- Status — сообщение содержит статус вызова из аспекта отправителя сообщения и причину его передачи;
- Notify — уведомление, содержащее информацию о вызове, например, индикацию временного прерывания (user suspend) или возобновления (user resume) вызова;
- Release Complete — полное разъединение вызова.

Протокол сигнализации RAS (Registration, Admission and Status — регистрация, подтверждение и статус) применяется для передачи служебных сообщений между терминалами и контроллером зоны. RAS-сообщения служат для регистрации терминалов, допуска их к сеансу связи, изменения используемой полосы пропускания, информирования о состоянии сеанса и его прекращении. В отсутствие контроллера зоны протокол RAS не используется.

Основные процедуры в рамках протокола RAS:

- обнаружение контроллера зоны;
- регистрация оконечного узла (терминала);
- управление доступом;
- управление шириной полосы пропускания;
- определение местонахождения оконечного узла;
- получения подробной статусной информации о вызовах.

Оконечные узлы используют протокол RAS для обнаружения контроллеров зоны, регистрации, а затем для получения разрешения на право использования части ресурсов системы, а также для получения

транспортных адресов других удалённых оконечных узлов. Контроллеры зоны, в свою очередь, посредством процедур регистрации и одобрения доступа используют протокол RAS для управления своей зоной, надзора за статусом регистрированных оконечных узлов, управления шириной полосы пропускания и определения местоположения оконечных узлов в других зонах посредством обмена адресной информацией с их контроллерами зоны.

8.2.8. Мультимедийная передача.

Протокол RTP (RFC 1889) обеспечивает в IP-сетях доставку адресатам аудио- и видеопотоков в масштабе реального времени. RTP идентифицирует тип и номер пакета, устанавливает в него метку синхронизации. На основе этой информации приёмный терминал синхронизирует звук, видео и данные, осуществляет их последовательное и непрерывное воспроизведение. Корректное функционирование RTP возможно при наличии в абонентских терминалах механизмов буферизации принимаемой информации.

Транспортный протокол управления передачей в режиме реального времени RTCP (RFC 1889) контролирует реализацию функций RTP. Он также отслеживает качество обслуживания и снабжает соответствующей информацией компоненты, участвующие в конференции.

8.2.9. Эволюция H.323

Первоначально протокол H.323 был предназначен исключительно для локальных сетей и не охватывал проблемы QoS и надёжности.

Вторая версия протокола H.323 одобрена в феврале 1998 г. Были введены некоторые новые функции, связанные с технологией VoIP.

Особое внимание уделялось механизмам обеспечения надёжной H.323-коммуникации в рамках рекомендации H.235[1]:

- подтверждение достоверности — механизм, которым подтверждается достоверность оконечных точек, участвующих в конференции;
- неприкосновенность данных — механизм контроля целостности принятых пакетных данных;
- защита персональной информации / конфиденциальность коммуникации путём кодирования и декодирования;
- невозможность отрицания — способ предотвращения возможности отрицания участия в конференции.

Другим улучшением в этой версии стало добавление процедуры ускоренного соединения (Fast Connect) — нового быстрого метода установления вызова, а также процедуры передачи сообщений H.245

[1] ITU-T Recommendation H.235.0, H.323 Security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems. 2005.

по каналу установления вызова (туннелирование), вследствие чего потребность надёжных (TCP) соединений в вызове была сведена к одному соединению, что дополнительно сократило время установления вызова.

Кроме того, во второй версии H.323 были введены первые дополнительные услуги в серии рекомендаций H.450: переадресация вызова (Call Transfer)[1] и изменение маршрута вызова (Call Diversion)[2].

Вторая версия охватывает и некоторые аспекты QoS, обеспечивая оконечным точкам H.323 возможность введения параметров качества для медиа потоков.

Улучшения второй версии коснулись и контроллера зоны — была введена концепция альтернативных контроллеров зоны, на которые перенаправлялась часть запросов с основного контроллера зоны в случае высокой нагрузки или неисправности.

Третья версия была утверждена в сентябре 1999 г. В этой версии стала возможной передача сигнализации для установления большего числа вызовов посредством одного TCP-соединения, а также удержание установленного TCP-соединения при отсутствии вызова.

Дополнение E (Annex E) к рекомендации H.323 обеспечило стандарту H.323 альтернативное решение — передачу сигнализации для установления вызова по ненадёжному каналу, используя протокол UDP, что позволило уменьшить время установления вызова и улучшить управление такими параметрами, как время повторной передачи неподтверждённых сообщений и определение ошибки удалённой H.323 оконечной точки, с которой осуществлена коммуникация.

В третьей версии было разработано дополнение рекомендации H.225.0, Annex G, описывающее методы и сигнализацию, необходимую для распознавания адреса, разрешения доступа, обмена информацией о тарификации и ценах на услуги, а также регистрации использования между административными доменами. Кроме того, это дополнение ввело в архитектуру H.323 новый элемент — разграничитель (Border Element).

В четвёртую версию, принятую в ноябре 2000 г., было введено множество улучшений с целью удержания в то время передовой позиции протокола VoIP. Улучшения коснулись надёжности, наращиваемости и гибкости структуры H.323.

Протокол для взаимодействия межсетевого шлюза (Media Gateway, MG) и контроллера межсетевого шлюза (Media Gateway Controller, MGC) разработан исследовательской группой 16 организации ITU- T в сотрудничестве с организацией IETF и описан в рекомендации H.248.

[1] ITU-T Recommendation H.450.2: Call transfer supplementary service for H.323. 1998.

[2] ITU-T Recommendation H.450.3: Call diversion supplementary service for H.323. 1998.

Помимо того, что в четвёртой версии была расширена группа поддерживаемых дополнительных услуг, также были введены два новых механизма предоставления дополнительных услуг — механизм управления H.323-устройствами, базирующийся на протоколе HTTP, и механизм управления, базирующийся на стимулировании. Механизмы описаны в дополнениях (Annex K и L) к рекомендации H.323.

Пятая версия H.323 была одобрена в июле 2003 г. и, в отличие от предыдущих версий, была направлена на стабилизацию протокола.

8.3. Концепция Softswitch. Протокол SIP

Softswitch является носителем интеллектуальных возможностей сети, который координирует управление обслуживанием вызовов, сигнализацию и функции, обеспечивающие установление соединения через одну или несколько сетей.

Softswitch: управляет обслуживанием вызовов; координирует обмен сигнальными сообщениями между сетями.

8.3.1. Архитектура Softswitch

Архитектура Softswitch (рис. 8.3) представляет собой набор функциональных объектов (функций, а не физических объектов), соединённых между собой посредством интерфейсов.

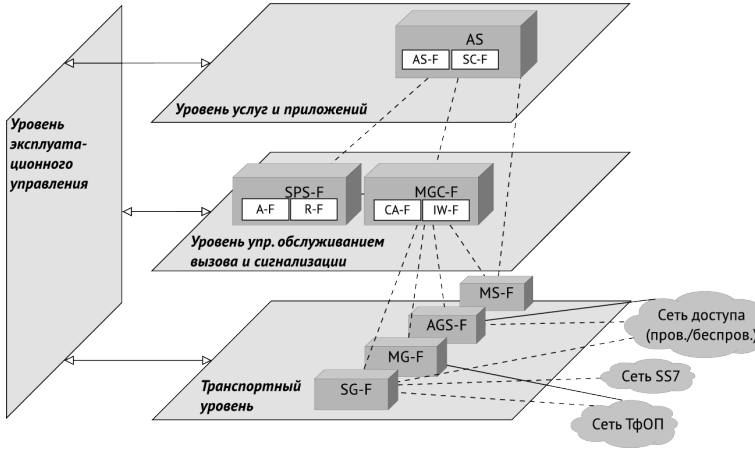


Рис. 8.3. Архитектура Softswitch (пунктирная линия — сигнализация, сплошная — данные)

В зависимости от своей функциональности функциональные объекты (ФО) распределены по функциональным уровням. Выделяют четыре функциональных уровня:

- *транспортный уровень (Transport Plane)* — отвечает за транспортировку сообщений по сети связи и обеспечивает доступ к сети IP-телефонии сигнальной и/или пользовательской информации, поступающей со стороны других сетей или терминалов;
- *уровень управления обслуживанием вызова и сигнализации (Call Control & Signaling Plane)* — управляет основными элементами сети IP-телефонии;
- *уровень услуг и приложений (Service & Application Plane)* — реализует управление услугами и/или приложениями в сети IP-телефонии, их логику и выполнение, а также управление специализированными компонентами передачи пользовательской информации (например, медиасерверами);
- *уровень эксплуатационного управления (Management Plane)* — обеспечивает выполнение функций активизации абонентов и услуг, техобслуживания, биллинга и пр.

Элементы транспортного уровня:

- *ФО шлюза сигнализации (Signaling Gateway Function, SG-F)* — обеспечивает обмен сигнальной информацией между сетью IP-телефонии и ТфОП или между транзитной пакетной IP-сетью и сетью сотовой подвижной связи с коммутацией каналов на базе стека SS7; использует протоколы Sigtran типов TUA, SUA и M3UA over SCTP;
- *ФО сигнализации шлюза доступа (Access Gateway Signaling Function, AGS-F)* — обеспечивает обмен сигнальной информацией между сетью IP-телефонии и сетью доступа с коммутацией каналов на базе интерфейса V5.1/V5.2 или ISDN, а также между транзитной сетью подвижной связи с коммутацией пакетов и сетью сотовой подвижной связи на базе TDM или ATM; использует протоколы Sigtran типов TUA, SUA и M3UA over SCTP;
- *ФО медиашлюза (Media Gateway Function, MG-F)* — обеспечивает сопряжение IP-сети с портом доступа, с соединительной линией или с совокупностью портов и/или соединительных линий, выполняя таким образом функции шлюза между пакетной сетью и внешними сетями с коммутацией каналов, такими как ТфОП, сеть сотовой подвижной связи или ATM; использует протоколы и технологии RTP/RTCP, TDM, H. 248 и MGCP;
- *ФО медиасервера (Media Server Function, MS-F)* — обеспечивает управление обработкой пользовательского пакетного трафика от приложений; использует протоколы SIP, MGCP и H. 248.

Элементы уровня управления обслуживанием вызова и сигнализации:

- *ФО контроллера медиашлюзов (Media Gateway Controller Function, MGC-F)* — представляет собой логический элемент управления обслуживанием вызова и сигнализации для одного или более транспортных шлюзов:
  - *ФО устройства управления шлюзом (Call Agent Function, CA-F)* — обеспечивает обработку вызова и определяет состояние процесса его обслуживания; может использовать протоколы SIP, SIP-T, BICC, H. 323, Q. 931, Q. SIG, INAP, ISUP, TCAP, BSSAP, RANAP, MAP и CAP;
  - *ФО взаимодействия (Interworking Function, IW-F)* — обеспечивает взаимодействие между разными сетями сигнализации (например, IP и ATM, OKC7 и SIP/H.323 и т. п.);
- *ФО SIP-прокси-сервера (SIP Proxy Server Function, SPS-F)*:
  - *ФО маршрутизации (Routing Function, R-F)* — предоставляет информацию о маршрутизации вызова ФО MGC-F; может использовать протоколы ENUM и TRIP;
  - *ФО учёта стоимости (Accounting Function, A-F)* — собирает учётную информацию о вызовах для целей биллинга, а также обеспечивает аутентификацию, идентификацию и учёт в удалённых сетях; может использовать протоколы RADIUS и AuC.

Элементы уровня услуг и приложений:

- *ФО сервера приложений (Application Server Function, AS-F)* — обеспечивает выполнение услуг для одного или более приложений; использует протоколы SIP, MGCP, H. 248, LDAP, HTTP, CPL и XML;
- *ФО управления услугами (Service Control Function, SC-F)* — обеспечивает управление логикой услуг; использует протоколы INAP, CAP и MAR, открытые API типа JAIN и Parlay.

Физически элементы сети могут выполнять как одну, так и несколько функций, а также одна функция может быть распределена между несколькими элементами сети. Так, в модуле контроллера медиашлюзов могут быть реализованы MGC-F, CA-F, IW-F, R-F/A-F, SPS-F и др.

8.3.2. Протоколы в сетях Softswitch

Протокол MGCP

Протокол управления медиашлюзом (Media Gateway Control Protocol,
MGCP) является внутренним протоколом для обмена информацией между функциональными блоками распределённого шлюза. Перенос сообщений протокола MGCP обеспечивает протокол UDP.

Для описания процесса обслуживания вызова с использованием протокола MGCP разработана модель организации соединения, в основу

которой положены два компонента: *оконечная точка или устройство (Endpoints)* и *подключение (Connections)*.

Оконечная точка — это порт оборудования, являющегося источником или приёмником информации. Порт может быть физическим или виртуальным. Каждый порт определяется идентификатором, содержащим доменное имя шлюза и локальное имя в шлюзе.

Соединение — подключение порта к одному из двух концов соединения, которое создаётся между ним и другим портом. Соединение может связывать порты разных шлюзов через сеть с IP-маршрутизацией или порты внутри одного шлюза.

При установлении, поддержании и разрушении соединения устройство управления и шлюз обмениваются командами и ответами, которые представляют собой набор текстовых строк.

Команды состоят из следующих компонент: *кода команды, идентификатора транзакции, идентификатора порта, версии протокола.*

В протоколе MGCP определены следующие команды:
- CreateConnection (CRCX) — создать соединение;
- ModifyConnection (MDCX) — модифицировать соединение;
- DeledeConnection (DLCX) — завершить соединение;
- Notify (NTFY) — уведомить;
- NotificationRequest (RQNT) — запрос уведомления;
- EndpointConfiguration (EPCF) — конфигурация портов;
- AuditEndpoint (AUEP) — проверить порт;
- AuditConnection (AUCX) — проверить соединение;
- ReStartInPrgress (RSIP) — рестарт.

Ответы состоят из следующих компонент: *кода ответа, идентификатора транзакции, комментария, параметров* (обязательных и не обязательных).

Определены следующие *основные параметры*:
- CallId (C) — идентификатор сеанса связи;
- ConnectionId (I) — идентификатор подключения;
- Mode (M) — режим соединения;
- RequestedInfo (F) — запрашиваемая информация;
- ResponseAck (K) — подтверждение транзакции;
- BearerInformation (B) — закон кодирования;
- RequestIdentifier (X) — идентификатор запроса;
- LocalConnectionOptions (L) — параметры порта;
- RequestedEvents (R) — запрашиваемые события;
- SignalRequests (S) — требование передать сигнал;
- NotifiedEntity (N) — уведомляемый объект;
- DigitMap (D) — план нумерации;
- QuarantineHandling (Q) — карантинная обработка;
- DetectEvents (T) — выявляемые события;

– ConnectionParameters (P) — параметры соединения;
– RestartMethod (RM) — метод рестарта;
– ReasonCode (E) — код причины;
– RestartDelay (RD) — задержка рестарта;
– ObservedEvents (O) — обнаруженные события;
– LocalConnectionDescriptor (LCD) — локальные параметры соединения на передающей стороне;
– RemoteConnectionDescriptor (RCD) — удалённые параметры соединения на приёмной стороне.

Протокол Megaco/H.248

Для переноса сигнальных сообщений Megaco/H.248 могут использоваться протоколы UDP, TCP, SCTP или технология ATM.

Для описания процесса обслуживания вызова с использованием протокола Megaco разработана модель организации соединения, в основу которой положены два компонента: *порт (Termination)* и *контекст (Context)*.

Порты являются источниками и приёмниками речевой информации и могут быть физическими (аналоговые телефонные интерфейсы оборудования) или виртуальными (существующие только в течение разговорной сессии).

Контекст — это абстрактное представление соединения двух или более портов одного шлюза. Контекст имеет уникальный идентификатор.

При помощи протокола Megaco/H.248 контроллер может изменять свойства портов шлюза. Свойства портов группируются в дескрипторы, которые включаются в команды управления портами.

Megaco/H.248 определяет восемь команд, которые обеспечивают возможность управления и манипулирования контекстами и окончаниями:

– Add — добавить окончание к контексту;
– Modify — изменить свойства окончания;
– Subtract — удалить окончание из контекста;
– Move — переместить окончание из одного контекста в другой;
– AuditValue — определить текущее состояние окончания;
– AuditCapabilities — определить состояния, которые может принимать окончание;
– Notify — уведомить о событиях, которые произошли в транспортном шлюзе;
– ServiceChange —уведомить об изменении обслуживания.

Megaco/H.248 определяет ряд дескрипторов, предназначенных для использования вместе с командами и ответами:

- дескриптор модема — специфицирует тип модема и связанные с ним параметры, которые следует использовать в соединениях модема при передаче аудио, видео или данных;
- дескриптор мультиплексирования — характеризует тип мультиплексирования в мультимедийном терминале;
- дескриптор среды — описывает различные информационные потоки (медиапотоки);
- дескрипторы потока — используются между MG и Softswitch для указания, какие медиапотоки взаимосвязаны;
- дескриптор среды — описывает различные информационные потоки (медиапотоки);
- дескрипторы LocalDescriptor и RemoteDescriptor — содержат или не содержат несколько описаний сеансов SDR, определяющих сеанс на локальном и удалённом концах соединения соответственно;
- дескриптор событий — содержит RequestIdentifier и список событий, которые MG должен обнаруживать;
- дескриптор сигналов — содержит список сигналов, которые должно подавать оконечное оборудование;
- дескриптор проверки — задаёт перечень информации, которую необходимо передавать из MG в Softswitch;
- дескриптор ServiceChangeDescriptor — используется только в сочетании с командой ServiceChange и включает в себя тип изменения обслуживания, причину изменения обслуживания и новый адрес для использования после изменения обслуживания;
- дескриптор DigitMap — описывает план нумерации;
- дескриптор StatisticsDescriptor — содержит информацию, которая относится к использованию оконечного оборудования в данном контексте;
- дескриптор ObservedEvents — используется для информирования Softswitch об обнаруженных событиях;
- дескриптор Error — передаётся в ответе, когда не может быть выполнена команда.

Команды могут группироваться в *транзакции*, причём в одной транзакции могут быть команды, относящиеся к разным контекстам. После приёма транзакции получатель последовательно выполняет команды, вложенные в неё.

Несколько транзакций могут передаваться по сети в виде *сообщений*, снабжённых заголовком, идентифицирующим отправителя. *Идентификатором сообщения (Message Identifier, MID)* служит назначенное имя (например, адрес в домене, имя в домене, имя устройства) объекта, передающего сообщение. Транзакции в пределах сообщения обрабатываются в произвольном порядке. Сообщения Megaco/H.248 по сути являются только транспортным механизмом.

Протокол Megaco/H.248 определяет типовые наборы характеристик, сигналов и событий для Softswitch и шлюзов разных типов,

чтобы обеспечить возможность их взаимодействия. Типовой набор характеризуется базовым описанием, свойствами, предусматриваемыми событиями, поддерживаемыми сигналами, предоставляемыми статистическими данными, любыми процедурами, относящимися к надлежащей поддержке набора. Он содержит следующие разделы:

- *Package* — содержит общее описание набора, определяющее его имя, идентификатор, текстовое описание, версию и опциональные поля;
- *Properties* — определяет свойства (характеристики) набора и содержит имя каждого свойства, его идентификатор, текстовое описание, тип, возможные значения, специфицирующие свойство и характеристики;
- *Events* — определяет событие и содержит имя события, его идентификатор, текстовое описание, параметры дескриптора Events и параметры дескриптора ObservedEvents;
- *Signals* — определяет сигналы, имя и идентификатор каждого сигнала, его текстовое описание, тип, продолжительность, дополнительные параметры;
- *Statistics* — определяет статистические данные, содержит имя и идентификатор данных каждого вида, их текстовое описание, единицы измерения;
- *Procedures* определяют дополнительные аспекты использования набора.

Протокол SIP

Протокол инициирования сеансов (Session Initiation Protocol, SIP) разработан группой MMUSIC (Multiparty Multimedia Session Control) комитета IETF[1] и используется для организации, модификации и завершения сеансов связи. Протокол SIP не принимает непосредственного участия в передаче голосовых, видео и других данных, а лишь отвечает за установление связи.

В основу протокола рабочая группа MMUSIC заложила следующие принципы:

- персональная мобильность пользователей — услуги связи предоставляются вне зависимости от местонахождения пользователя;
- масштабируемость сети;
- расширяемость протокола — возможно дополнение протокола новыми функциями при введении новых услуг и его адаптации к работе с различными приложениями.

[1]SIP: Session Initiation Protocol, RFC 2543 / M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg; SIP: Session Initiation Protocol, RFC 3261 / J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler.

Кроме того, протокол SIP поддерживает преобразование имён, переадресацию, маршрутизацию, идентификацию и аутентификацию пользователя при его перемещении из одного места в другое.

В сети на базе SIP определены следующие элементы:

- *клиент UAC (User Agent Client)* — инициирует SIP-запросы;
- *сервер UAS (User Agent Server)* — принимает запросы и передаёт обратно ответы:
  - *прокси-сервер (Proxy Server)* — обрабатывает запросы пользователя;
  - *сервер переадресации (Redirect Server)* — предназначен для определения текущего адреса вызываемого пользователя;
  - *сервер регистрации местоположения (Registrars / Location Server)* — позволяют агентам регистрировать своё местоположение, реализуя тем самым услуги мобильности.

Все сообщения SIP делятся на запросы клиента серверу и ответы сервера клиенту. Сообщения SIP могут переноситься как протоколом TCP, так и протоколом UDP. Все сообщения SIP представляют собой последовательности текстовых строк, структура и синтаксис которых соответствуют протоколу HTTP:

- *стартовая строка* — представляет собой начальную строку любого SIP-сообщения и содержит в случае запроса *тип запроса, текущий адрес узла-адресата, номер версии протокола*, а в случае ответа — *номер версии протокола, тип ответа, короткую расшифровку ответа*;
- *заголовки сообщений* — содержат информацию, необходимую для обработки сообщения:
  - *общие заголовки*: Call-ID (идентификатор соединения), Contact (контакт), CSeq (порядковый номер запроса/ответа), Date (дата), Encryption (кодирование), From (источник запроса), To (адресат), Via (путь), Record-Route (запись маршрута);
  - *заголовки содержания* — переносят информацию о размере тела сообщения или об источнике запроса;
  - *заголовки с дополнительной информацией о запросе*: Accept (принимается), Accept-Encoding (кодирование принимается), Accept-Language (язык поддерживается), Authorization (авторизация), Hide (скрыть), Max-Forwards (максимальное количество переадресаций), Organization (организация), Priority (приоритет), Proxy-Authorization (авторизация прокси-сервера), Proxy-Require (требование прокси-сервера), Route (маршрут), Response-Key (ключ кодирования ответа), Subject (тема), User-Agent (агент пользователя);
  - *заголовки с дополнительной информацией об ответе*: Allow (разрешение), Proxy-Authenticate (подтверждение подлинности прокси-сервера), Retry-After (повторить через некоторое вре-

мя), Server (сервер), Unsupported (не поддерживается), Warning (предупреждение), WWW-Authenticate (аутентификация WWW-сервера);

- *тело сообщения* — содержит запросы (команды) SIP:
  - *INVITE* — приглашает пользователя принять участие в сеансе связи, и обычно содержит описание сеанса связи, вид принимаемой информации и параметры, необходимые для приёма информации;
  - *ACK* — подтверждает приём ответа на команду INVITE, содержит описание сеанса связи, переданное вызывающим пользователем;
  - *CANCEL* — отменяет обработку ранее переданных запросов;
  - *BYE* — разрушает соединение;
  - *REGISTER* — сообщает текущее местоположение пользователя;
  - *OPTIONS* — содержит информацию о возможностях терминального оборудования вызываемого пользователя;
  - *INFO* — используется для переноса между шлюзами сигнальных сообщений в течение сеанса связи, для переноса сигналов DTMF, созданных в ходе сеанса, для переноса информации об остатке на счете (билинговой информации), для переноса между участниками сеанса связи изображений и другой не потоковой информации;
  - *SUBSCRIBE* — подписка на предоставление информации о состоянии определённого ресурса;
  - *MESSAGE* — предназначен для реализации служб интерактивного обмена текстовыми сообщениями с использованием модели, аналогичной отправке SMS.

Для организации взаимодействия с существующими приложениями IP-сетей и обеспечения мобильности пользователей протокол SIP использует адрес, подобный адресу электронной почты. В качестве адресов рабочих станций используются SIP URL, которые бывают четырёх типов: `имя@домен`, `имя@IP-адрес`, `имя@хост`, `№телефона@шлюз`. Первая часть адреса идентифицирует пользователя, зарегистрированного в домене или на рабочей станции, а вторая часть — устройство или домен.

8.3.3. Протокол SDP

Протокол описания сессий (Session Description Protocol, SDP)[1] содержит механизм описания характеристик сеанса — время проведения, требуемые ресурсы и т.д. В SDP предусмотрена возможность изменения параметров сеансов в оперативном режиме.

SDP содержит следующие данные:

[1] *Handley M., Jacobson V.* SDP: Session Description Protocol, RFC 2327.

– информацию о медиапотоках;
– адреса назначения медиапотоков;
– номера UDP портов для отправителя и получателя;
– типы потока;
– медиаформаты, которые могут использоваться во время сессии;
– время начала, завершения и повторов сессии;
– информацию об инициаторе широковещательной сессии.

Описание сессии SDP:

– поле *Версия протокола* содержит версию протокола SDP;
– поле *Владелец / создатель и идентификатор сессии* служит глобальным идентификатором версии описания сессии:
  – username — идентифицирует пользователя;
  – session id — уникальный идентификатор сессии;
  – version — номер версии данного объявления;
  – network type — тип сети (например, «IN» — Интернет);
  – address type — тип адреса (например, «IP4» или «IP6»);
  – address — глобальный уникальный адрес хоста, с которого была создана данная сессия;
– поле *Имя сессии* указывает имя сессии;
– поле *Информация о сессии* может использоваться для определения медиапотока;
– поле *URI описания сессии* указывает на дополнительную информацию о конференции (сессии);
– поле *e-mail адрес*;
– поле *Телефонный номер*;
– поле *Информация о соединении* содержит данные о соединении:
  – network type — тип сети (например, «IN» — Интернет);
  – address type — тип адреса (например, «IP4» или «IP6»);
  – connection address — адрес соединения;
– поле *Информация о ширине полосы пропускания* определяет желаемую ширину полосы пропускания, которая должна использоваться сессией и медиапотоком;
– поле *Время* определяет время начала и конца сессии;
– поле *Интервалы повторения сессий*;
– поле *Объявление временной зоны* определяет сдвиги времени по отношению к базовому времени повторов сессий;
– поле *Криптографический ключ*;
– поле *Атрибуты сессии* могут быть определены как атрибуты «уровня сессии», атрибуты «уровня медиа»;
– поля *Имя медиа* и *Адрес транспорта*:
  – media — содержит тип медиапотока;
  – port — транспортный порт, в который будет передаваться медиапоток;
  – transport — транспортный протокол;
  – fmt list — форматы медиа.

8.3.4. Услуги в сетях Softswitch

Архитектура Softswitch даёт возможность операторам и/или провайдерам услуг предоставлять услуги, реализованные в виде приложений как от производителя Softswitch, так и от сторонних производителей, а также самостоятельно разрабатывать свои собственные приложения. Это возможно благодаря основанным на открытых стандартах прикладным программным интерфейсам API:

- Parlay — платформа для разработки, интеграции и развёртывания приложений на базе технологии Java;
- JAIN (Java Advanced Intelligent Network) — сетевая топология на базе Java, позволяющая осуществлять интеграцию протоколов IP и IN, обеспечивающая переносимость услуг, конвергенцию сетей и защищённый доступ как к телефонным сетям, так и к сетям передачи данных;
- CORBA (Common Object Request Broker Architecture) — открытая, независимая от поставщиков архитектура и инфраструктура, которую используют прикладные вычислительные системы для обеспечения их совместной работы в компьютерных сетях;
- XML (Extensible Markup Language) — язык разметки, который рассматривается как стандартный способ обмена информацией в средах, не использующих общие платформы;
- CPL (Call Processing Language) — язык, который может быть использован для описания и управления услугами IP-телефонии;
- CGI (Common Gateway Interface) — стандарт интерфейса, используемого для связи внешней программы с веб-сервером;
- сервисные Java-приложения.

8.4. Концепция IMS

Концепция *IMS (IP Multimedia Subsystem)* была предложена 3GPP в начале 2003 г. Эта концепция определяет сетевую архитектуру, которая опирается на пакетную транспортную сеть и обеспечивает управление сеансами связи и доставку в рамках этих сеансов любых типов информации — речи, данных, видео, мультимедиа. Следует заметить, что в системах, отвечающих концепции IMS, услуги могут предоставляться разными сервис-провайдерами и доставляться до пользователей по различным (проводным и беспроводным) сетям доступа.

Концепция IMS была стандартизована в спецификациях 3GPP R.5. Позднее к разработке спецификаций и стандартов IMS присоединились другие организации: 3GPP2, занимающаяся разработками для сетей CDMA2000, ETSI, группа Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), работающая в области конвергенции фиксированных сетей. Альянс

Open Mobile Alliance (OMA) определил приложения и услуги, работающие поверх IMS, а Internet Engineering Task Force (IETF) — протоколы сетевого уровня. ETSI, отраслевые группы Форума мультисервисной коммутации (Multiservice Switching Forum, MSF) и Альянса для продвижения решений для телекоммуникационной отрасли (Alliance for Telecommunications Industry Solutions, ATIS) одобрили IMS в качестве основы сетевой инфраструктуры следующего поколения.

8.4.1. Архитектура IMS

Архитектура IMS (рис. 8.4) [27] представляет собой набор функций, соединённых стандартными интерфейсами (табл. 8.3). Физически элементы сети могут выполнять как одну, так и несколько функций, а также одна функция может быть распределена между несколькими элементами сети.

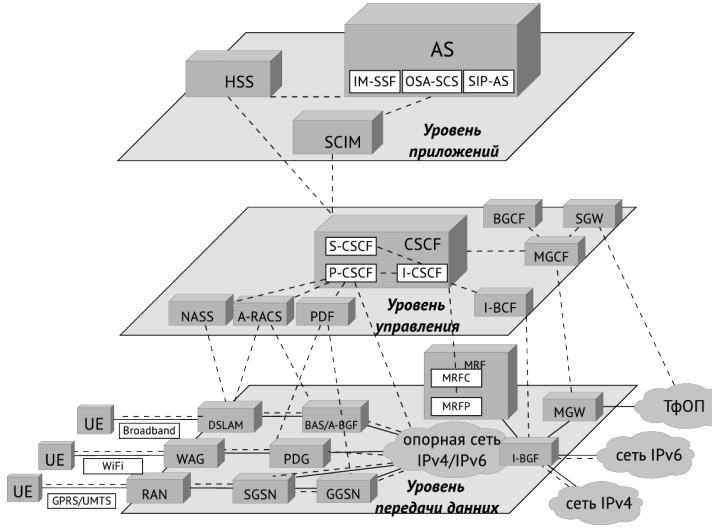


Рис. 8.4. Архитектура IMS (пунктирная линия — сигнализация, сплошная — данные)

Таблица 8.3

Описание стандартных интерфейсов

Название интерфейса	**Элементы IMS**	**Описание**	**Протокол**
Cr	MRFC, AS	Используется MRFC для передачи данных (скриптов и др/ ресурсов) от AS	HTTP поверх TCP/SCTP
Cx	I-CSCF, S-CSCF, HSS	Используется для взаимодействия между I-CSCF/S-CSCF и HSS	Diameter
Dh	SIP AS, OSA, SCF, IM-SSF, HSS	Используется AS для поиска нужного HSS	Diameter
Dx	I-CSCF, S-CSCF, SLF	Используется I-CSCF/S-CSCF для поиска правильного HSS	Diameter
Gm	UE, P-CSCF	Используется для обмена сообщениями между UE и CSCF	SIP
Go	PDF, GGSN	Даёт возможность операторам управлять QoS на уровне передачи данных и обмениваться информацией между IMS и GPRS сетями	COPS (Rel5), Diameter (Rel6+)
Gq	P-CSCF, PDF	Используется для обмена политиками между P-CSCF и PDF	Diameter
ISC	S-CSCF, I-CSCF, AS	Используется для обмена сообщениями между CSCF и AS	SIP
Ma	I-CSCF -> AS	Используется для прямого перенаправления SIP-запросов, предназначенных серверам приложений (AS)	SIP
Mg	MGCF -> I-CSCF	MGCF преобразует сигнализацию ISUP в сигнализацию SIP и перенаправляет её в I-CSCF	SIP
Mi	S-CSCF -> BGCF	Используется для обмена сообщениями между S-CSCF и BGCF	SIP
Mj	BGCF -> MGCF	Используется для обмена сообщениями между BGCF и MGCF в некоторых сетях IMS	SIP
Mk	BGCF -> BGCF	Используется для прямого обмена сообщениями между BGCFs и IMS	SIP

Таблица 8.3

Описание стандартных интерфейсов (продолжение)

Название интерфейса	**Элементы IMS**	**Описание**	**Протокол**
Mm	I-CSCF, S-CSCF, IP-сеть	Используется для обмена сообщениями между IMS и IP-сетями	-
Mn	MGCF, IM-MGW	Даёт возможность управлять ресурсами уровня передачи данных	H.248
Mp	MRFC, MRFP	Используется для обмена сообщениями между MRFC и MRFP	H.248
Mr	S-CSCF, MRFC	Используется для обмена сообщениями между S-CSCF и MRFC	SIP
Mw	P-CSCF, I-CSCF, S-CSCF	Используется для обмена сообщениями между несколькими CSCF	SIP
Rf	P-CSCF, I-CSCF, S-CSCF, BGCF, MRFC, MGCF, AS	Используется для offline-обмена информацией с CCF	Diameter
Ro	AS, MRFC	Используется для online-обмена информацией с ECF	Diameter
Sh	SIP AS, OSA SCS, HSS	Используется для обмена сообщениями между SIP AS/OSA SCS и HSS	Diameter
Si	IM-SSF, HSS	Используется для обмена сообщениями между IM-SSF и HSS	MAP
Sr	MRFC, AS	Используется MRFC для передачи документов (скриптов и др. ресурсов) для AS	HTTP
Ut	UE, AS (SIP AS, OSA SCS, IM-SSF)	Разрешает UE управлять информацией, касающейся его сервисов	HTTP(s)

Выделяют три уровня:

– *пользовательский уровень или уровень передачи данных (User Plane)* — отвечает за подключение абонентов к инфраструктуре IMS;
– *уровень управления (Control Plane)* — отвечает за все действия по управлению сеансами связи (регистрирует абонентские устройства и направляет сигнальные сообщения протокола SIP к соответствующим серверам приложений);

- *уровень приложений (Application Plane)* — обеспечивает обслуживание конечных пользователей.

Элементы уровня передачи данных:

- *функция обеспечения мультимедийных ресурсов (Media Resource Function, MRF)*:
  - *процессор мультимедийных ресурсов (MRF Processor, MRFP)* — обеспечивает обработку мультимедийных данных;
  - *контроллер мультимедийных ресурсов (MRF Controller, MRFC)* — обеспечивает реализацию услуг конференц-связи, оповещения или перекодирования передаваемого сигнала посредством управления MRFP при помощи протоколов сигнализации;
- *медиа-шлюз (Media Gateway, MGW)* — обеспечивает прямое и обратное преобразование потоков сетей с коммутацией пакетов в потоки сетей с коммутацией каналов;
- *функция межсетевого пограничного шлюза (Interconnect Border Gateway Function, I-BGF)* — обеспечивает взаимодействие между сетями IPv4 и IPv6, отвечает за обеспечение функций безопасности (трансляция адресов и портов NAPT, функции firewall, инструменты QoS);
- *шлюзовой узел GPRS (Gateway GPRS Support Node, GGSN)* — обеспечивает взаимодействие сети сотовой связи и инфраструктуры IMS;
- узел обслуживания абонентов GPRS (Serving GPRS Support Node, SGSN) — обеспечивает обработку данных абонентов GPRS;
- *сети радиодоступа (Radio Access Network, RAN)* — обеспечивают взаимодействие сотовых систем электросвязи и инфраструктуры IMS;
- *шлюз пакетной передачи данных (Packet Data Gateway, PDG)* — обеспечивает доступ пользовательского оборудования WLAN к инфраструктуре IMS, а именно ретранслирует IP-адреса, регистрирует пользовательское оборудование в IMS, обеспечивает выполнение функций безопасности;
- *шлюз беспроводного доступа (Wireless Access Gateway, WAG)* — обеспечивает соединение сетей WLAN и IMS;
- *функция пограничного шлюза доступа для широкополосного пользовательского оборудования (Access Border Gateway Function / Broadband Access Switch, A-BGF/BAS)* — обеспечивает доступ широкополосного пользовательского оборудования к инфраструктуре IMS;
- *цифровой абонентский шлюз доступа (Digital Subscriber Line Access Multiplexer, DSLAM)* — обеспечивает соединение абонентов, использующих широкополосный доступ к инфраструктуре IMS.

Элементы уровня управления:

- *функция управления вызовами и сеансами (Call Session Control Function, CSCF)* — обеспечивает доставку услуг реального времени посредством транспорта IP:
  - *обслуживающая CSCF (Serving CSCF, S-CSCF)* — обрабатывает все SIP-сообщения, которыми обмениваются оконечные устройства;
  - *прокси CSCF (Proxy CSCF, P-CSCF)* — обеспечивает обработку запросов от терминалов IMS к другим элементам IMS, а также выполняет ряд требований, относящихся к обеспечению безопасности (аутентификацию пользователя, контроль за корректностью передаваемых сигнальных сообщений, сбор данных о предоставленных пользователю сервисах);
  - *запрашивающая CSCF (Interrogating CSCF, I-CSCF)* — назначает S-CSCF для конкретного абонента, определяет привилегии абонента по доступу к услугам;
- *функция управления шлюзами (Breakout Gateway Control Function, BGCF)* — управляет маршрутизацией вызовов между сетью с коммутацией каналов (ТфОП или GSM) и сетью IMS;
- *функция управления медиа-шлюзами (Media GatewaysControl Function,*
  MGCF) — управляет соединениями в транспортных шлюзах IMS, используя H.248/ MEGACO;
- *шлюз сигнализации (Signaling Gateway, SGW)* — обеспечивает преобразование сигнализации ТфОП в вид, понятный MGCF;
- *подсистема управления ресурсами и доступом (Resource and Access Control, RACS)* — обеспечивает функции управления доступом в сеть, управление преобразованием сетевых адресов и портов, присвоение приоритета;
- *функция выбора политики (Policy Decision Function, PDF)* — определяет возможность организации сеанса или его запрета, необходимость изменения параметров сеанса и т.д.;
- *подсистема подключения сети (Network Attachment Subsystem, NASS)* — осуществляет динамическое назначение IP-адресов, аутентификацию на IP-уровне, авторизацию доступа к сети, управление местонахождением на IP-уровне.

Элементы уровня приложений:

- *элемент управления взаимодействием возможных услуг (Service Capability Interaction Manager, SCIM)* — обеспечивает управление взаимодействием плоскости приложений и ядра IMS;
- *SIP-сервер приложений (SIP Application Server, SIP AS)* — обеспечивает выполнение услуг на базе SIP;
- *сервер возможных услуг, базирующихся на открытом доступе к услугам (Open Service Access — Service Capability Server, OSA-SCS)* — обеспечивает доступ к услугам посредством стандартного

программного интерфейса приложений;
- *сервер коммутации услуг (IP Multimedia – Service Switching Function, IM-SSF)* — служит для взаимодействия подсистемы IMS с услугами, разработанными для системы мобильной связи GSM;
- *сервер телефонных приложений (Telephony Application Server, TAS)* — принимает и обрабатывает сообщения протокола SIP, обеспечивает базовые сервисы обработки вызовов (включая анализ цифр, маршрутизацию, установление, ожидание и перенаправление вызовов, конференц-связь и т.д.), обеспечивает сервисную логику для обращения к медиасерверам при необходимости воспроизведения оповещений и сигналов прохождения вызова, отвечает за сигнализацию SIP к функции MGCF для выдачи команды медиашлюзам на преобразование битов речевого потока TDM (ТфОП) в поток IP RTP и направление его на IP-адрес соответствующего IP-телефона;
- *сервер домашних абонентов (Home Subscriber Server, HSS)* — обеспечивает открытый доступ в режиме чтения/записи к индивидуальным данным пользователя, связанным с услугами.

8.4.2. Услуги в сетях IMS

Для реализации новых конвергентных услуг с гарантией качества обслуживания сервисная архитектура сети должна отвечать следующим требованиям:
- отделение уровней транспорта и доступа от сервисного уровня (прозрачность доступа);
- управление сеансом связи, в ходе которого задействуются несколько сервисов связи реального времени;
- совместимость с имеющимися сервисами интеллектуальной сети (IN), к которым относятся: определение имени вызывающей стороны, бесплатный номер (800), переносимость локального номера, сервисы, соответствующие стандартам CAMEL, ANSI-41 и т. д.;
- прозрачное взаимодействие с телефонными сетями (планы нумерации, сигнализация прохождения вызовов);
- конвергенция проводных и беспроводных сервисов;
- объединение голосовых услуг с сервисами реального времени (обмен мгновенными сообщениями);
- стандартизованные механизмы обмена пользовательской информацией между сервисами;
- стандартизованные механизмы аутентификации и биллинга конечных пользователей;
- стандартизованный, общий для всех сервисов графический пользовательский интерфейс;
- открытые стандартные интерфейсы и API для новых сервисов, разработанные сервис-провайдерами и третьими фирмами.

В сетях IMS определены следующие услуги:

– услуги, основанные на информации о присутствии и доступности пользователя — позволяют обеспечить доставку информации «правильному» человеку и/или на «правильное» устройство, т.е. при помощи протокола SIP можно обеспечить «прозрачное» переключение, например, между сотовой, WiFi или наземной связью с помощью одного устройства;
– услуги, основанные на информации о местоположении пользователя — позволяют предоставить пользователю информацию, актуальную для него в данный момент (например, прогноз погоды, информация о дорожной ситуации и т.п.);
– единый механизм авторизации, не связанный с конкретным устройством или технологией;
– управление групповыми списками;
– групповое общение (Group Communication);
– Push-To-Talk — услуга, работающая в полудуплексном (half-duplex) режиме, когда сотовый телефон используется как терминал системы профессиональной мобильной радиосвязи (основное преимущество — возможность «группового вызова», т.е. общения по принципу «один–многие»);
– Push-To-Show;
– доска для записей (Whiteboard) — услуга, позволяющая двум или нескольким абонентам совместно редактировать рисунки и документы в режиме реального времени. Все, что делается одним участником сеанса, видят в режиме on-line все остальные участники;
– многопользовательские игры в реальном времени (шахматы и другие игры);
– голосовые вызовы с усовершенствованными функциями (Enriched Voice Calling) — включают видеотелефонию и возможность добавления к вызовам своего контента;
– совместное использование файлов в сети (File Sharing);
– обеспечение необходимого уровня безопасности.

8.4.3. Протокол SIP

Протокол SIP предназначен для управления сеансами связи (инициация, модификация, завершение). Использование SIP в IMS позволяет реализовать услугу конференц-связи, поскольку любое число абонентов может динамически подключаться к сеансу и выходить из него. Кроме того, SIP даёт возможность динамически в рамках существующего сеанса связи подключать новые услуги (например, сеанс связи можно начать с текстового чата, потом добавить голосовую связь, а затем при необходимости и видео). Наконец, средства SIP способны при инициации или модификации сеанса связи учитывать характеристики канала доступа и терминала каждого пользователя и задействовать их оптимальным образом.

8.4.4. Преимущества и недостатки IMS

IMS обладает следующими преимуществами:
- предоставление множества услуг — нет жёсткой привязки средств управления услугами и способа их доставки до абонента с самими услугами, внедрение принципиально нового сервиса не требует построения соответствующей инфраструктуры для его доставки;
- хорошая масштабируемость сети оператора — модернизировать инфраструктуру сети можно поэлементно (например, при увеличении объёма трафика можно модернизировать только элементы уровня передачи данных, а при увеличении числа абонентов — элементы уровня управления);
- независимость IMS от специфики сетевого транспорта и каналов доступа делает её хорошей основой для конвергенции служб фиксированной и мобильной связи.

Недостатки IMS:
- для полноценного перехода к IMS операторам связи необходимо выстроить новую схему управления сеансами связи и модернизировать системы поддержки эксплуатации и бизнес-операций, а также обеспечить поддержку маршрутизаторами протокола IPv6;
- отсутствие терминалов, ориентированных на работу в IMS-сетях, — оконечное оборудование должно уметь инициировать и обрабатывать IMS-запросы, поддерживать работу сложных приложений;
- отсутствие поддержки non-SIP-приложений в рамках SIP-ориентированной архитектуры IMS.

8.5. Концепция A-IMS

В июле 2006 г. рабочая группа, в которую вошли ведущие поставщики телекоммуникационного оборудования Lucent Technologies, Cicso Systems, Motorola, Nortel и Qualcomm, под руководством оператора мобильной связи Verizon Wireless объявила о создании архитектуры *Advances to IMS (A-IMS)* [28—30].

Архитектура A-IMS [28; 30] является дальнейшим развитием стандарта IMS и призвана преодолеть его недостатки. Одной из проблем IMS является отсутствие поддержки non-SIP-приложений в рамках имеющейся SIP-ориентированной архитектуры IMS. Архитектура A-IMS позволяет осуществлять взаимодействие между SIP и non-SIP-приложениями, обеспечивает более полный policy-контроль над ними и управление сетевыми ресурсами, отвечающими за QoS, мобильность, безопасность, доступ и т.п. Включённые в архитектуру дополнения и усовершенствования применимы для построения сетей связи на основе разных технологий доступа (3G, xDSL, WiMax, Cable) или конвергентных VoIP-сетей.

Основные элементы A-IMS:

- *подсистема управления приложениями (Application Manager, AM)* — элемент управления SIP-сессиями, выполняющий функции P-CSCF, I-CSCF, S-CSCF, BGCF;
- *подсистема управления данными об услугах (Services Data Manager, SDM)* — осуществляет хранение данных как для SIP, так и для non-SIP-приложений, включает в себя функциональность HSS и AAA, а также (опционально) SLF (Subscriber Location Function), KMF (Key Management Function) и Accounting;
- *подсистема управления несущей (Bearer Manager, BM)* — осуществляет контроль на уровне транспортного потока (несущей): контролирует применение соответствующих политик, правил, осуществляет управление потоками данных PFO (Packet Flow Optimization), идентификацию вторжений;
- *подсистема управления безопасностью (Security Manager, SM)* — выполняет задачи мониторинга событий в сети, обнаружения аномалий на основе программных алгоритмов, управления элементами сети для отражения угроз, управления IDS/IDP и политиками безопасности;
- *подсистема управления политиками (Policy Manager, PM)* — обеспечивает общее управление и контроль над распределением ресурсов сети (QoS, PFO, mobility, access и т.п.); поддерживает как SIP, так и non-SIP-приложения.

Дополнительные элементы A-IMS:

- *терминал доступа (Access Terminal, AT)* — оконечное устройство (фиксированное или мобильным), имеющее возможность предоставить доступ пользователей к услугам с помощью разных технологий (xDSL,WiFi, EVDO и т.п.);
- *шлюз IP (IP Gateway, IPGW)* — поддерживает взаимодействие между канальным и сетевым уровнями сети передачи данных, осуществляет аутентификацию устройств и переадресацию, отвечает за подсчёт пакетного трафика и обеспечение QoS;
- *посредник при предоставлении услуг (Service Broker, SB)* — представляет собой один из компонентов, отвечающих за механизм вызова (запуска) приложения с разных платформ (как использующих, так и не использующих SIP), при этом хранит логику предоставления услуг и управляет взаимодействием различных приложений на уровне сессий, являясь главным связующим звеном между SIP и non-SIP-приложениями;
- *функция управления ключами (Key Management Function, KMF)* — хранит ключи (абонентские и сетевые), которые используются при аутентификации абонентских устройств;
- *Regulatory and PSTN Servers* — обеспечивают интерфейс для выполнения определённых задач перехвата вызовов и сбора информации для компетентных ведомств.

Глава 9. Лабораторный практикум

В основу лабораторного практикума положены материалы из источников [13; 31].

9.1. Лабораторная работа. Методы кодирования и модуляция сигналов

9.1.1. Цели работы

Изучение методов кодирования и модуляции сигналов с помощью высокоуровнего языка программирования Octave. Определение спектра и параметров сигнала. Демонстрация принципов модуляции сигнала на примере аналоговой амплитудной модуляции. Исследование свойства самосинхронизации сигнала.

9.1.2. Теоретические сведения

Основы работы в Octave

Octave — высокоуровневый интерпретируемый язык программирования, предназначенный для решения задач вычислительной математики.

Интерпретатор Octave запускается из терминала операционной системы с помощью команды `octave` для работы с помощью консоли или `qtoctave` для работы с оконным интерфейсом.

В окне интерпретатора Octave пользователь может вводить как отдельные команды языка Octave, так и группы команд, объединяемые в программы. Если строка заканчивается символом «;», то результаты на экран не выводятся. Если в конце строки символ «;» отсутствует, то результаты работы выводятся на экран. Текст в стоке, который идет после символа %, является строкой комментария и интерпретатором не обрабатывается.

Octave имеет два режима работы: *терминальный* и *программный*. В терминальном режиме отдельные команды последовательно вводятся в окне интерпретатора. В программном режиме создается текстовый файл (с расширением **.m**), в котором хранятся последовательно выполняемые команды, впоследствии запускаемые на выполнение в среде Octave.

Простейшие арифметические операции в Octave:

`+` — сложение;
`-` — вычитание;
`*` — умножение;

`/` — деление;
`^` — возведение в степень.

Для определения переменной необходимо набрать *имя переменной*, символ «=» и *значение переменной*, где знак равенства – это оператор присваивания:

```
имя переменной = значение выражения
```

Система различает большие и малые буквы в именах переменных. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением. Если речь идет о символьной или строковой переменной, то выражение в правой части оператора присваивания следует брать в одинарные кавычки.

Если команда не содержит знака присваивания, то по умолчанию вычисленное значение присваивается специальной системной переменной `ans`. Причем полученное значение можно использовать в последующих вычислениях, но важно помнить, что значение `ans` изменяется после каждого вызова команды без оператора присваивания.

Системные переменные:
- `ans` – результат последней операции без знака присваивания;
- `i`, `j` – мнимая единица ($\sqrt{-1}$);
- `pi` – число π (3.141592653589793);
- `e` – число e (экспонента 2.71828183);
- `inf` – машинный символ бесконечности (∞);
- `NaN` – неопределенный результат.

Все перечисленные переменные можно использовать в математических выражениях.

Команда `clear` предназначена для уничтожения определения одной или нескольких переменных:

```
clear имя переменной
```

В общем виде обращение к функции в Octave имеет вид

```
имя переменной = имя функции(аргумент)
```

или

```
имя функции(аргумент)
```

Если имя переменной указано, то ей будет присвоен результат работы функции. Если же оно отсутствует, то значение вычисленного функцией результата присваивается системной переменной `ans`.

Примеры работы с тригонометрическими функциями:

```
>>>x=pi/2;   % Определение значения аргумента
>>>y=sin(x)  % Вызов функции
y = 1
>>>cos(pi/3) % Вызов функции
ans = 0.50000
```

Здесь `>>>` — знак приглашения Octave для ввода команд.

Некоторые встроенные функции Octave:
- `sin(x)` — синус числа x;

– `cos(x)` — косинус числа x;
– `tan(x)` — тангенс числа x;
– `exp(x)` — экспонента числа x;
– `log(x)` — натуральный логарифм числа x;
– `round(x)` — обычное округление числа x до ближайшего целого;
– `rem(x,y)` — вычисление остатка от деления x на y;
– `sign(x)` — сигнум-функция числа x, выдаёт 0, если $= 0$, -1 — при $x < 0$ и 1 при $x > 0$;
– `sqrt(x)` — корень квадратный из числа x;
– `abs(x)` — модуль числа x.

Операции отношения выполняют сравнение двух операндов и определяют, истинно выражение или ложно:

`<` — меньше;
`>` — больше;
`= =` — равно;
`~=` —не равно;
`<=` — меньше или равно;
`>=` — больше или равно.

Синтаксис функции, определяемой пользователем:

```
function name1 [,name2,...] = fun(var1 [,var2,...])
```

Здесь `name1 [,name2,...]` — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений, `fun` — имя функции, `var1 [,var2,...]` — входные параметры.

Все имена переменных внутри функции, а также имена из списка входных и выходных параметров воспринимаются системой как локальные, т.е. эти переменные считаются определенными только внутри функции. Программы и функции в Octave могут быть созданы при помощи текстового редактора и сохранены в виде файла с расширением `.m` или `.M`. Но при создании и сохранении функции следует помнить, что ее имя должно совпадать с именем файла. Программу можно запустить на выполнение, указав имя файла, в котором она сохранена. Обращение к функции осуществляется так же, как и к любой другой встроенной функции системы, то есть с указанием входных и выходных параметров. Вызвать функцию можно из командной строки или использовать ее как один из операторов программы.

Массив — множественный тип данных, состоящий из фиксированного числа элементов одного типа. Как и любой другой переменной, массиву должно быть присвоено имя.

Самый простой способ задать одномерный массив в Octave имеет вид

```
имя массива = Xn:dX:Xk
```

Здесь `Xn` — значение первого элемента массива, `Xk` — значение последнего элемента массива, `dX` — шаг, с помощью которого формируется каждый следующий элемент массива, т.е. значение второго элемента составит `Xn+dX`, третьего `Xn+dX+dX` и так далее до `Xk`.

Примеры создания массивов:

```
>>>A=1:5
A =
1 2 3 4 5
>>>B=2:2:10
B =
2 4 6 8 10
>>>xn=-3.5;xk=3.5;dx=0.5;
>>>X=xn:dx:xk
X =
Columns 1 through 8:
-3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0
Columns 9 through 15:
0.5 1.0 1.5 2.0 2.5 3.0 3.5
```

Обратиться к элементу вектора можно, указав имя массива и порядковый номер элемента в круглых скобках:

```
>>>x=[2 4 6 8 10];
>>>y=[-1.2 3.4 -0.8 9.1 5.6 -7.3];
>>>x(1) % значение первого элемента массива x
ans = 2
>>>y(5) % значение пятого элемента массива y
ans = 5.6000
>>>x(1)/2+y(3)^2-x(4)/y(5)
ans = 0.21143
```

Ввод элементов матрицы также осуществляется в квадратных скобках, при этом элементы строки отделяются друг от друга пробелом или запятой, а строки разделяются между собой точкой с запятой. Обратиться к элементу матрицы можно, указав после имени матрицы, в круглых скобках, через запятую, номер строки и номер столбца, на пересечении которых элемент расположен:

```
>>>M=[2 4 6;1 3 5;7 8 9]
M =
2 4 6
1 3 5
7 8 9
>>>M(1,2)
ans = 4
>>>M(3,1)
ans = 7
```

Построение графиков в Octave

Для того чтобы построить двумерный график функции $f(x)$, необходимо сформировать два массива x и y одинаковой размерности, а затем обратиться к функции `plot`.

Синтаксис функции `plot`:

`plot(x1,y1,s1,x2,y2,s2,...,xn,yn,sn)`
Здесь `x1,x2,...,xn` — массивы абсцисс графиков; `y1,y2,...,yn` — массивы ординат графиков; `s1,s2,...,sn` — строка форматов, определяющая параметры линии и, при необходимости, позволяющая вывести легенду.

В строке форматов могут участвовать символы, отвечающие за тип линии, маркер и его размер, цвет линии и вывод легенды. За сплошную линию отвечает символ «-». Цвет линии определяется буквой латинского алфавита: `y` — жёлтый, `m` — розовый, `c` — голубой, `r` — красный, `g` — зелёный, `b` — синий, `w` — белый. Некоторые символы маркера: `.` — точка, `*` — звёздочка, `x` — крестик, `+` — плюс, `o` — незакрашенный круг, `p` — незакрашенный квадрат.

Например, для построения графика функции $y = \sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x$ на интервале $[-10; 10]$ (рис. 9.1) можно использовать следующий листинг:

```
% Формирование массива x:
x=-10:0.1:10;
% Формирование массива y.
y=sin(x)+1/3*sin(3*x)+1/5*sin(5*x);
% Построение графика функции:
plot(x,y, "-ok; y=sin(x)+ (1/3)*sin(3*x)+(1/5)*sin(5*x);",
  "markersize",4)
% Отображение сетки на графике
grid on;
% Подпись оси X:
xlabel('x');
% Подпись оси Y:
ylabel('y');
Название графика:
title('y=sin x+ (1/3)sin(3x)+(1/5)sin(5x)');
% Экспорт рисунка в файл .eps:
print ("plot-sin.eps", "-mono", "-FArial:16", "-deps")
```

Если повторно обратиться к функции `plot`, то в этом же окне будет стёрт первый график и нарисован второй. Для построения нескольких графиков в одной системе координат можно поступить одним из следующих способов:

1) обратиться к функции `plot` следующим образом:
   `plot(x1,y1,x2,y2,... xn,yn),`
   где `x1`, `y1` — массивы абсцисс и ординат первого графика, `x2`, `y2` — массивы абсцисс и ординат второго графика, ..., `xn`, `yn` — массивы абсцисс и ординат n-го графика;
2) каждый график изображать с помощью функции `plot(x,y)`, но перед обращением к функциям `plot(x2,y2)`, `plot(x3,y3)`, ...,

`plot(xn,yn)` вызвать команду `hold on`, которая блокирует режим очистки окна.

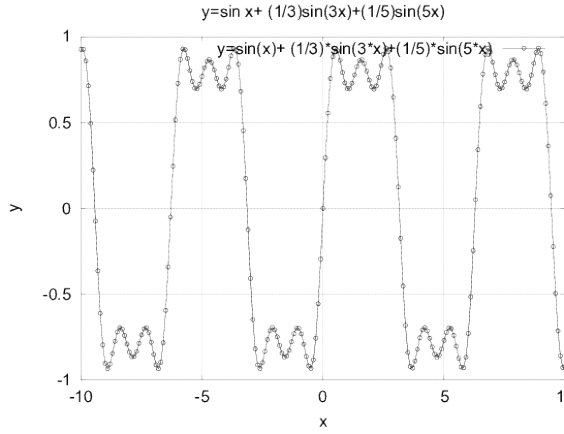


Рис. 9.1. График функции $y = \sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x$ **на интервале** $[-10; 10]$

9.1.3. Порядок выполнения работы

1. Ознакомиться с теоретическим материалом главы 3.
2. Выполнить задания разделов 9.1.3–9.1.3. Получить соответствующие графики.
3. Составить отчёт о выполненной работе, включив в него листинги программ и полученные графики.

Разложение импульсного сигнала в частичный ряд Фурье

В цифровой технике основным типом сигналов является импульсный сигнал. Импульсный сигнал можно описать математически в синусоидальной форме. Такой тип сигнала называется меандром.

Меандр — бесконечный, периодический сигнал прямоугольной формы (импульсный сигнал), широко используемый в радиотехнике. Длительность импульса и длительность паузы в периоде такого сигнала равны.

Спектр меандра имеет вид

$$s(t) = \frac{A}{2} + \frac{2A}{\pi}\left(\cos\left(\frac{2\pi}{T}t\right) - \frac{1}{3}\cos\left(3\frac{2\pi}{T}t\right) + \frac{1}{5}\cos\left(5\frac{2\pi}{T}t\right) - \ldots\right).$$

Гармоники, образующие меандр, имеют амплитуду, обратно пропорциональную номеру соответствующей гармоники.

Задание: разработать код m-файла, результатом выполнения которого являются графики меандра (рис. 9.2), реализованные с различным количеством гармоник.

Листинг программы в Octave:

```
% meandr.m
% количество отсчетов:
N=8;
% частота дискретизации:
t=-1:0.01:1;
% значение амплитуды:
A=1;
T=1;
nh=(1:N)*2-1;
% входной сигнал:
harmonics=cos(2*pi*nh'*t/T);
Am=2/pi./nh;
Am(2:2:end)=-Am(2:2:end);
s1=harmonics.*repmat(Am',1,length(t));
s2=cumsum(s1);
for k=1:N
    subplot(4,2,k)
    plot(t, s2(k,:))
end
```

Здесь функция `repmat(A,M,N)` формирует массив из частей; имеет три входных аргумента: массив `A`, количество строк `M` и столбцов `N` для вновь создаваемого массива; `cumsum` — суммирование элементов массива с накоплением.

Определение спектра и параметров сигнала

Задание. Определить спектр двух отдельных сигналов и их суммы.

Частота дискретизации (количество отсчётов) выбирается на основе теоремы Котельникова как удвоенная ширина спектра исходного сигнала (таким образом, в следующем примере достаточно было взять частоту дискретизации 80 Гц).

Попробуйте выполнить задание с другой частотой дискретизации. Что будет, если взять частоту дискретизации меньше 80 Гц?

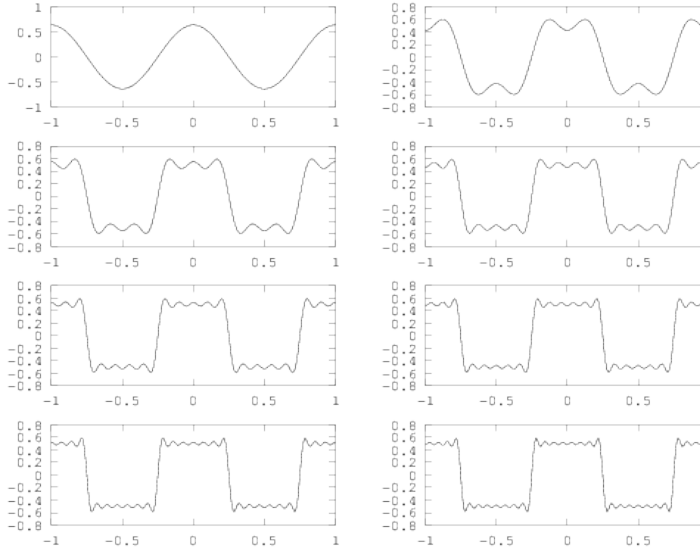


Рис. 9.2. Графики меандра, содержащего различное число гармоник

Для двух синусоидальных сигналов (рис. 9.3) требуется определить их спектр. В файле spectre.m задаем параметры сигналов:

```
% spectre.m
mkdir 'signal';
mkdir 'spectre';
tmax = 0.5;% Длина сигнала (с)
fd = 512; % Частота дискретизации (Гц) (количество отсчётов)
f1 = 10;% Частота первого сигнала (Гц)
f2 = 40;% Частота второго сигнала (Гц)
a1 = 1;% Амплитуда первого сигнала
a2 = 0.7;% Амплитуда второго сигнала
fd2 = fd/2; % Спектр сигнала
% Рассмотрим два сигнала (синусоиды) разной частоты
t = 0:1./fd:tmax; % Массив отсчётов времени
signal1 = a1*sin(2*pi*t*f1);
signal2 = a2*sin(2*pi*t*f2);
```

```
plot(signal1,'b'); % голубая
hold on
plot(signal2,'r'); % красная
hold off
title('Signal');
print 'signal/spectre.png';
```

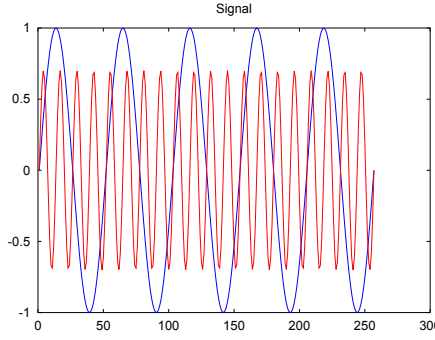


Рис. 9.3. Два синусоидальных сигнала разной частоты

С помощью быстрого преобразования Фурье найдем спектры сигналов (рис. 9.4), добавив в файл `spectre.m` следующий код.

```
% Посчитаем спектр
% Амплитуды преобразования Фурье  сигнала 1
spectre1 = abs(fft(signal1,fd));
% Амплитуды преобразования Фурье сигнала 2
spectre2 = abs(fft(signal2,fd));
% Построение спектров сигналов
plot(spectre1,'b'); % голубая
hold on
plot(spectre2,'r'); % красная
hold off
title('Spectre');
print 'spectre/spectre.png';
```

Учитывая реализацию преобразования Фурье, скорректируем график спектра (рис. 9.5): отбрасываются дублирующие отрицательные частоты, а также учитывается то, что на каждом шаге вычисления быстрого преобразования Фурье происходит суммирование амплитуд сигналов. Добавляем в файл `spectre.m` следующий код.

```
% Исправление графика спектра
f = 1000*(0:fd2)./(2*fd); % Сетка частот
```

```
% Нормировка спектров по амплитуде
spectre1 = 2*spectre1/fd2;
spectre2 = 2*spectre2/fd2;
% Построение спектров сигналов
plot(f,spectre1(1:fd2+1),'b'); % голубая
hold on
plot(f,spectre2(1:fd2+1),'r'); % красная
hold off
xlim([0 100]);
title('Fixed spectre');
xlabel('Frequency (Hz)');
print 'spectre/spectre_fix.png';
```

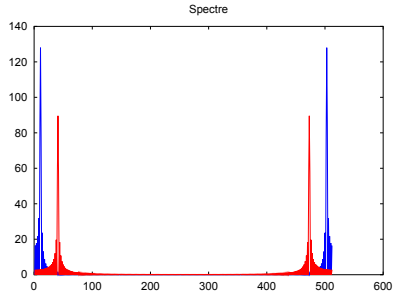


Рис. 9.4. График спектров синусоидальных сигналов

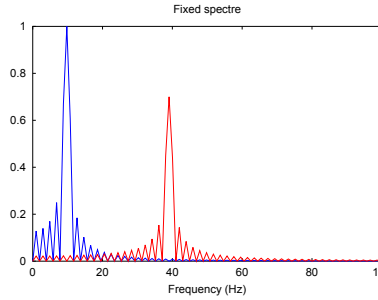


Рис. 9.5. Исправленный график спектров синусоидальных сигналов

Аналогично найдем спектр суммы рассмотренных сигналов (9.6), создав файл `spectre_sum.m` со следующим кодом.

```
% spectre_sum.m
mkdir 'signal';
mkdir 'spectre';
% Длина сигнала (с)
tmax = 0.5;
% Частота дискретизации (Гц) (количество отсчётов)
fd = 512;
% Частота первого сигнала (Гц)
f1 = 10;
% Частота второго сигнала (Гц)
f2 = 40;
% Амплитуда первого сигнала
a1 = 1;
% Амплитуда второго сигнала
```

```
a2 = 0.7;
% Спектр сигнала
fd2 = fd/2;
% Сумма двух сигналов (синусоиды) разной частоты
% Массив отсчётов времени:
t = 0:1./fd:tmax;
signal1 = a1*sin(2*pi*t*f1);
signal2 = a2*sin(2*pi*t*f2);
signal = signal1 + signal2;
plot(signal);
title('Signal');
print 'signal/spectre_sum.png';
% Подсчет спектра:
% Амплитуды преобразования Фурье сигнала
spectre = fft(signal,fd);
% Сетка частот
f = 1000*(0:fd2)./(2*fd);
% Нормировка спектра по амплитуде:
spectre = 2*sqrt(spectre.*conj(spectre))./fd2;
% Построение спектра сигнала
plot(f,spectre(1:fd2+1))
xlim([0 100]);
title('Spectre');
xlabel('Frequency (Hz)');
print 'spectre/spectre_sum.png';
```

В результате получим аналогичный предыдущему результат (рис. 9.7), т.е. спектр суммы сигналов равен сумме спектров сигналов, что вытекает из свойств преобразования Фурье.

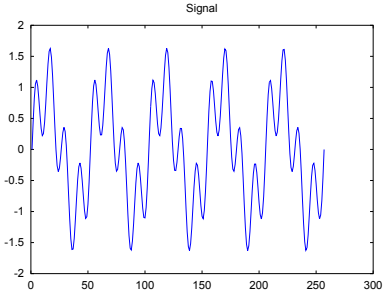


Рис. 9.6. Суммарный сигнал

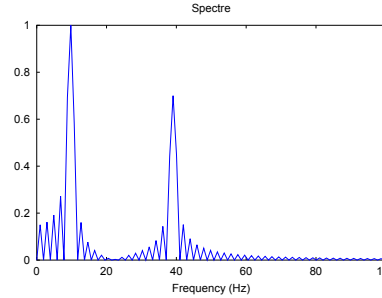


Рис. 9.7. Спектр суммарного сигнала

Демонстрация принципов модуляции сигнала на примере аналоговой амплитудной модуляции

Следующий код в файле `am.m` демонстрирует принципы модуляции сигнала на примере аналоговой амплитудной модуляции (рис.9.8).

```
% am.m
mkdir 'signal';
mkdir 'spectre';

% Модуляция синусоид с частотами 50 и 5
% Длина сигнала (с)
tmax = 0.5;
% Частота дискретизации (Гц) (количество отсчётов)
fd = 512;
% Частота сигнала (Гц)
f1 = 5;
% Частота несущей (Гц)
f2 = 50;
% Спектр сигнала
fd2 = fd/2;

% Построение графиков двух сигналов (синусоиды)
% разной частоты
% Массив отсчётов времени:
t = 0:1./fd:tmax;
signal1 = sin(2*pi*t*f1);
signal2 = sin(2*pi*t*f2);
signal = signal1 .* signal2;
plot(signal, 'b');
hold on

% Построение огибающей:
plot(signal1, 'r');
plot(-signal1, 'r');
hold off
title('Signal');
print 'signal/am.png';

% Расчет спектра:
% Амплитуды преобразования Фурье-сигнала
spectre = fft(signal,fd);
% Сетка частот
f = 1000*(0:fd2)./(2*fd);

% Нормировка спектра по амплитуде:
spectre = 2*sqrt(spectre.*conj(spectre))./fd2;
```

```
% Построение спектра:
plot(f,spectre(1:fd2+1), 'b')
xlim([0 100]);
title('Spectre');
xlabel('Frequency (Hz)');
print 'spectre/am.png';
```

В результате получаем, что спектр произведения есть свёртка спектров (рис. 9.9).

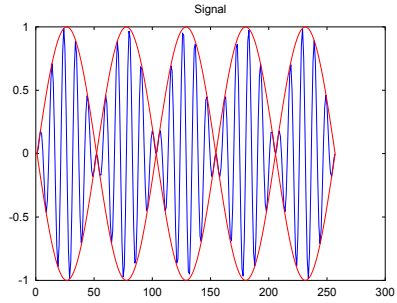


Рис. 9.8. Сигнал и огибающая при амплитудной модуляции

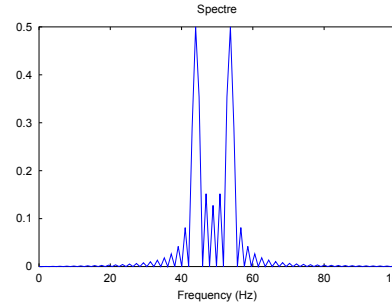


Рис. 9.9. Спектр сигнала при амплитудной модуляции

Кодирование сигнала. Исследование свойства самосинхронизации сигнала

По заданной выходной битовой последовательности требуется получить кодированный сигнал для нескольких кодов, проверить свойства самосинхронизуемости кода.

Создаём файл `main.m`:

```
% main.m
pkg load signal;

% Задаем входную кодовую последовательность:
data=[0 1 0 0 1 1 0 0 0 1 1 0];

% Задаем входную кодовую последовательность
% для проверки свойства самосинхронизации:
data_sync=[0 0 0 0 0 0 0 1 1 1 1 1 1 1];

% Построение графиков кодированного сигнала
mkdir 'signal';
```

```
axis("auto");

% Униполярное кодирование
wave=unipolar(data);
plot(wave);
ylim([-1 6]);
title('Unipolar');
print 'signal/unipolar.png';

% Кодирование ami
wave=ami(data);
plot(wave)
title('AMI');
print 'signal/ami.png';

% Кодирование NRZ
wave=bipolarnrz(data);
plot(wave);
title('Bipolar Non-Return to Zero');
print 'signal/bipolarnrz.png';

% Кодирование RZ
wave=bipolarrz(data);
plot(wave)
title('Bipolar Return to Zero');
print 'signal/bipolarrz.png';

% Манчестерское кодирование
wave=manchester(data);
plot(wave)
title('Manchester');
print 'signal/manchester.png';

% Дифференциальное манчестерское кодирование
wave=diffmanc(data);
plot(wave)
title('Differential Manchester');
print 'signal/diffmanc.png';

% Построение графиков кодированного сигнала
% для проверки свойства самосинхронизации
mkdir 'sync';
axis("auto");

% Униполярное кодирование
wave=unipolar(data_sync);
```

```
plot(wave);
ylim([-1 6]);
title('Unipolar');
print 'sync/unipolar.png';

% Кодирование AMI
wave=ami(data_sync);
plot(wave)
title('AMI');
print 'sync/ami.png';

% Кодирование NRZ
wave=bipolarnrz(data_sync);
plot(wave);
title('Bipolar Non-Return to Zero');
print 'sync/bipolarnrz.png';

% Кодирование RZ
wave=bipolarrz(data_sync);
plot(wave)
title('Bipolar Return to Zero');
print 'sync/bipolarrz.png';

% Манчестерское кодирование
wave=manchester(data_sync);
plot(wave)
title('Manchester');
print 'sync/manchester.png';

% Дифференциальное манчестерское кодирование
wave=diffmanc(data_sync);
plot(wave)
title('Differential Manchester');
print 'sync/diffmanc.png';
```

Следующая функция (в отдельном файле `maptowave.m`) по входному битовому потоку строит график сигнала:

```
% maptowave.m
function wave=maptowave(data)
        data=upsample(data,100);
        wave=filter(5*ones(1,100),1,data);
```

Каждая функция преобразования кодовой последовательности находится в отдельном файле. Например, униполярное кодирование реализуется с помощью следующей функции:

```
% unipolar.m
function wave=unipolar(data)
```

```
        wave=maptowave(data);
```

Кодирование AMI реализуется с помощью следующей функции:

```
% ami.m
function wave=ami(data)
        am=mod(1:length(data(data==1)),2);
        am(am==0)=-1;
        data(data==1)=am;
        wave=maptowave(data);
```

Кодирование NRZ реализуется с помощью следующей функции:

```
% bipolarnrz.m
function wave=bipolarnrz(data)
        data(data==0)=-1;
        wave=maptowave(data);
```

Кодирование RZ реализуется с помощью следующей функции:

```
% bipolarrz.m
function wave=bipolarrz(data)
       data(data==0)=-1;
       data=upsample(data,2);
       wave=maptowave(data);
```

Манчестерское кодирование реализуется с помощью следующей функции:

```
% manchester.m
function wave=manchester(data)
       data(data==0)=-1;
       data=upsample(data,2);
       data=filter([-1 1],1,data);
       wave=maptowave(data);
```

Дифференциальное манчестерское кодирование реализуется с помощью следующей функции:

```
% diffmanc.m
function wave=diffmanc(data)
        data=filter(1,[1 1],data);
        data=mod(data,2);
        wave=manchester(data);
```

Для построения спектра сигнала реализуем следующую функцию (в отдельном файле `calcspectre.m`):

```
% calcspectre.m
function spectre = calcspectre(wave)
        Fd = 512; % Частота дискретизации (Гц)
        Fd2 = Fd/2;
        Fd3 = Fd/2 + 1;
        X = fft(wave,Fd);
        spectre = X.*conj(X)/Fd;
        f = 1000*(0:Fd2)/Fd;
        plot(f,spectre(1:Fd3));
        xlabel('Frequency (Hz)');
```

и в файл `main.m` добавляем следующий код:

```
% Построение спектра сигнала
mkdir 'spectre';
axis("auto");
data_spectre=[0 1 0 1 0 1 0 1 0 1 0 1 0 1];

% униполярное кодирование
wave=unipolar(data_spectre);
spectre=calcspectre(wave);
title('Unipolar');
print 'spectre/unipolar.png';

% кодирование AMI
wave=ami(data_spectre);
spectre=calcspectre(wave);
title('AMI');
print 'spectre/ami.png';

% кодирование NRZ
wave=bipolarnrz(data_spectre);
spectre=calcspectre(wave);
title('Bipolar Non-Return to Zero');
print 'spectre/bipolarnrz.png';

% кодирование RZ
wave=bipolarrz(data_spectre);
spectre=calcspectre(wave);
title('Bipolar Return to Zero');
print 'spectre/bipolarrz.png';

% манчестерское кодирование
wave=manchester(data_spectre);
spectre=calcspectre(wave);
title('Manchester');
print 'spectre/manchester.png';

% дифференциальное манчестерское кодирование
wave=diffmanc(data_spectre);
spectre=calcspectre(wave);
title('Differential Manchester');
print 'spectre/diffmanc.png';
```

Запускаем главный скрипт `main.m`. В каталоге `signal` получаем поведение кодированного сигнала (рис. 9.10–9.15).

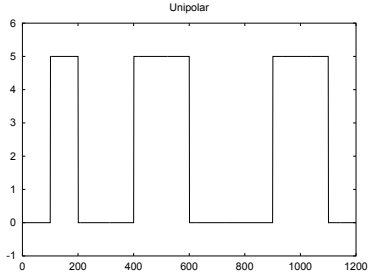


Рис. 9.10. Униполярное кодирование

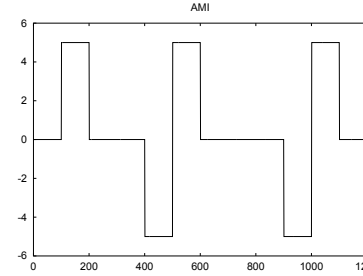


Рис. 9.11. Кодирование AMI

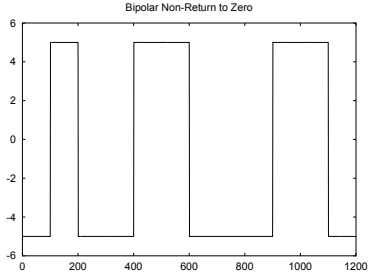


Рис. 9.12. Кодирование NRZ

Рис. 9.13. Кодирование RZ

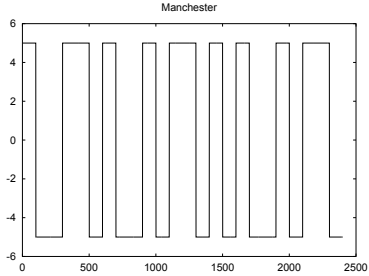


Рис. 9.14. Манчестерское кодирование

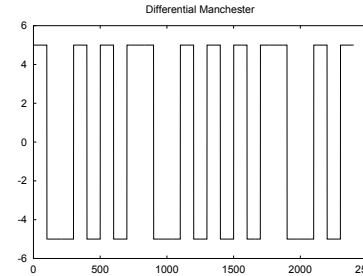


Рис. 9.15. Дифференциальное манчестерское кодирование

В каталоге `sync` иллюстрируются свойства самосинхронизации (рис. 9.16–9.21).

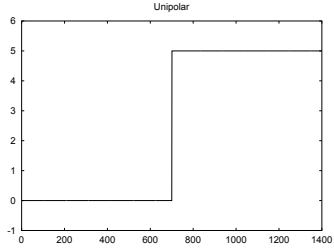


Рис. 9.16. Униполярное кодирование: нет самосинхронизации

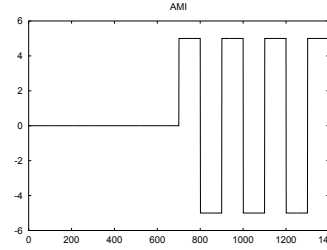


Рис. 9.17. Кодирование AMI: самосинхронизация при наличии сигнала

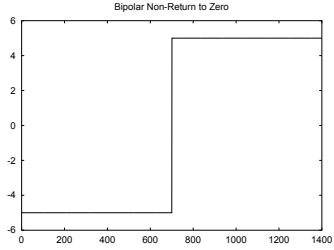


Рис. 9.18. Кодирование NRZ: нет самосинхронизации

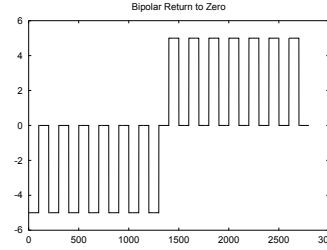


Рис. 9.19. Кодирование RZ: есть самосинхронизация

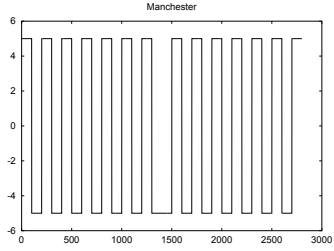


Рис. 9.20. Манчестерское кодирование: есть самосинхронизация

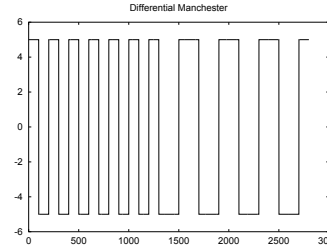


Рис. 9.21. Дифференциальное манчестерское кодирование: есть самосинхронизация

В каталоге `spectre` получаем спектр сигнала (рис. 9.22–9.27).

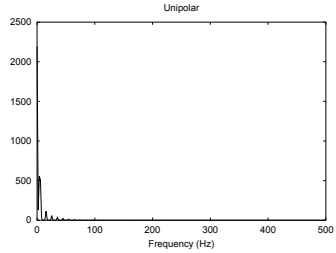


Рис. 9.22. Униполярное кодирование: спектр сигнала

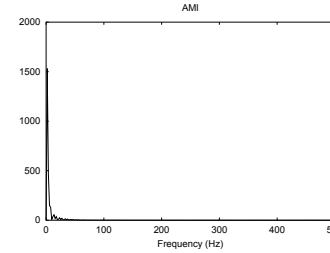


Рис. 9.23. Кодирование AMI: спектр сигнала

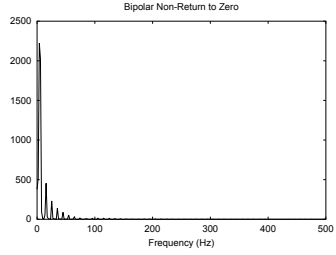


Рис. 9.24. Кодирование NRZ: спектр сигнала

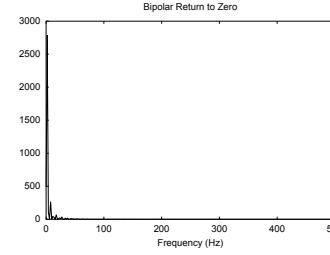


Рис. 9.25. Кодирование RZ: спектр сигнала

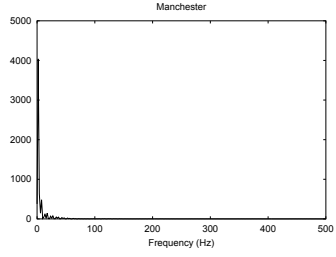


Рис. 9.26. Манчестерское кодирование: спектр сигнала

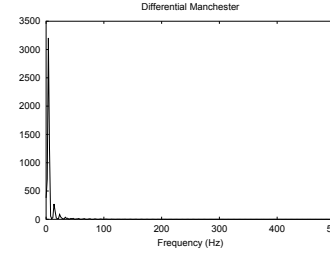


Рис. 9.27. Дифференциальное манчестерское кодирование: спектр сигнала

9.2. Лабораторная работа. Расчёт сети Fast Ethernet

9.2.1. Цели работы

Цель данной работы — изучение принципов технологий Ethernet и Fast Ethernet и практическое освоение методик оценки работоспособности сети, построенной на базе технологии Fast Ethernet.

9.2.2. Теоретические сведения

Технология Ethernet

Спецификация сети Ethernet была предложена фирмами DEC, Intel и Xerox (DIX) в 1980 г., и несколько позже на её основе появился стандарт IEEE 802.3.

Первые версии Ethernet v1.0 и Ethernet v2.0 в качестве среды передачи использовали только коаксиальный кабель. Стандарт IEEE 802.3 позволяет в качестве среды передачи использовать также витую пару и оптоволокно. В 1995 г. был принят стандарт IEEE 802.3u (Fast Ethernet) со скоростью 100 Мбит/с, а в 1997 г. — IEEE 802.3z (Gigabit Ethernet — 1000 Мбит/с). Осенью 1999 г. принят стандарт IEEE 802.3ab — Gigabit Ethernet на витой паре категории 5.

В обозначениях Ethernet (10BASE2, 100BASE-TX и др.) первый элемент обозначает скорость передачи данных в Мбит/с; второй элемент BASE означает, что используется прямая (немодулированная) передача; третий элемент обозначает округлённое значение длины кабеля в сотнях метров (10BASE2 – 185 м, 10BASE5 – 500 м) или тип среды передачи (T, TX, T2, T4 — витая пара; FX, FL, FB, SX и LX — оптоволокно; CX — твинаксиальный кабель для Gigabit Ethernet).

В основе Ethernet лежит *метод множественного доступа к среде передачи с прослушиванием несущей и обнаружением коллизий — CSMA/CD (Carrier Sense with Multiple Access and Collision Detection)*, реализуемый адаптерами каждого узла сети на аппаратном или микропрограммном уровне:

- все адаптеры имеют устройство доступа к среде (MAU) — трансивер, подключённый к общей (разделяемой) среде передачи данных;
- каждый адаптер узла перед передачей информации прослушивает линию до момента отсутствия сигнала (несущей);
- затем адаптер формирует кадр (frame), начинающийся с синхронизирующей преамбулы, за которой следует поток двоичных данных в самосинхронизирующемся (манчестерском) коде;
- другие узлы принимают посланный сигнал, синхронизируются по преамбуле и декодируют его в последовательность бит;

- окончание передачи кадра определяется обнаружением приёмником отсутствия несущей;
- в случае обнаружения *коллизии* (столкновения двух сигналов от разных узлов) передающие узлы прекращают передачу кадра, после чего через случайный промежуток времени (каждый через свой) осуществляют повторную попытку передачи после освобождения линии; при очередной неудаче делается следующая попытка (и так до 16 раз), причём интервал задержки увеличивается;
- коллизия обнаруживается приёмником по нестандартной длине кадра, которая не может быть меньше 64 байт, не считая преамбулы;
- между кадрами должен обеспечиваться временной зазор (*межкадровый или межпакетный промежуток, IPG – inter-packet gap*) длительностью 9,6 мкс — узел не имеет права начать передачу раньше, чем через интервал IPG, после определения момента пропадания несущей.

Определение 1. *Домен коллизий* — группа узлов, связанных общей средой (кабелями и повторителями) передачи.

Протяжённость домена коллизий ограничивается временем распространения сигнала между наиболее удалёнными друг от друга узлами.

Определение 2. *Диаметр домена коллизий* — расстояние между двумя наиболее удалёнными друг от друга оконечными устройствами.

Определение 3. *Битовый интервал* — время, необходимое для передачи одного бита.

Битовый интервал в Ethernet (при скорости 10 Мбит/с) составляет 0,1 мкс.

Технология Fast Ethernet

В технологии Fast Ethernet величина битового интервала составляет 0,01 мкс, что даёт десятикратное увеличение скорости передачи данных. При этом формат кадра, объём переносимых кадром данных и механизм доступа к каналу передачи данных остались без изменения по сравнению с Ethernet.

В Fast Ethernet используется среда передачи данных для работы на скорости 100 Мбит/с, которая в спецификации IEEE 802.3u имеет обозначения «100BASE-T4» и «100BASE-TX» (витая пара); «100BASE-FX» и «100BASE-SX» (оптоволокно).

Правила построения сети

Первая модель сети Fast Ethernet. Модель представляет собой, по сути, набор правил построения сети (табл. 9.1):

- длина каждого сегмента витой пары должна быть меньше 100 м;
- длина каждого оптоволоконного сегмента должна быть меньше 412 м;
- если используются кабели MII (Media Independent Interface), то каждый из них должен быть меньше 0,5 м;
- задержки, вносимые кабелем MII, не учитываются при оценке временных параметров сети, так как они являются составной частью задержек, вносимых оконечными устройствами (терминалами) и повторителями.

Таблица 9.1

Предельно допустимый диаметр домена коллизий в Fast Ethernet

Тип повторителя	Все сегменты TX или T4	Все сегменты FX	Сочетание сегментов (T4 и TX/FX)	Сочетание сегментов (TX и FX)
Сегмент, соединяющий два узла без повторителей	100	412,0	–	–
Один повторитель класса I	200	272,0	231,0	260,8
Один повторитель класса II	200	320,0	–	308,8
Два повторителя класса II	205	228,0	–	216,2

Стандартом определены два класса повторителей:

- повторители класса I выполняют преобразование входных сигналов в цифровой вид, а при передаче снова перекодируют цифровые данные в физические сигналы; преобразование сигналов в повторителе требует некоторого времени, поэтому в домене коллизий допускается только один повторитель класса I;
- повторители класса II немедленно передают полученные сигналы без всякого преобразования, поэтому к ним можно подключать только сегменты, использующие одинаковые способы кодирования

данных; можно использовать не более двух повторителей класса II в одном домене коллизий.

Вторая модель сети Fast Ethernet. Вторая модель содержит последовательность расчётов временных параметров сети при полудуплексном режиме обмена данными. Диаметр домена коллизий и количество сегментов в нём ограничены временем двойного оборота, необходимым для правильной работы механизма обнаружения и разрешения коллизий (табл. 9.2).

Таблица 9.2

Временные задержки компонентов сети Fast Ethernet

Компонент	Удельное время двойного оборота (би/м)	Максимальное время двойного оборота (би)
Пара терминалов TX/FX	–	100
Пара терминалов T4	–	138
Пара терминалов T4 и TX/FX	–	127
Витая пара категории 3	1,14	114 (100 м)
Витая пара категории 4	1,14	114 (100 м)
Витая пара категории 5	1,112	111,2 (100 м)
Экранированная витая пара	1,112	111,2 (100 м)
Оптоволокно	1,0	412 (412 м)
Повторитель класса I	–	140
Повторитель класса II, имеющий порты типа TX/FX	–	92
Повторитель класса II, имеющий порты типа T4	–	67

Время двойного оборота рассчитывается для наихудшего (в смысле распространения сигнала) пути между двумя узлами домена колли-

зий. Расчёт выполняется путём суммирования временных задержек в сегментах, повторителях и терминалах.

Для вычисления времени двойного оборота нужно умножить длину сегмента на величину удельного времени двойного оборота соответствующего сегмента. Определив времена двойного оборота для всех сегментов наихудшего пути, к ним нужно прибавить задержку, вносимую парой оконечных узлов и повторителями. Для учёта непредвиденных задержек к полученному результату рекомендуется добавить ещё 4 битовых интервала (би) и сравнить результат с числом 512. Если полученный результат не превышает 512 би, то сеть считается работоспособной.

Пример расчёта конфигурации сети Fast Ethernet

На рис. 9.28 приведён пример одной из предельно допустимых конфигураций сети Fast Ethernet.

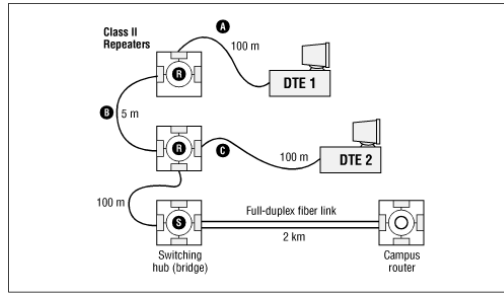


Рис. 9.28. Пример допустимой конфигурации сети Fast Ethernet

Диаметр домена коллизий вычисляется как сумма длин сегментов A (100 м), B (5 м) и C (100 м) и равен 205 м. Длина сегмента, соединяющего повторители, может быть более 5 м, если при этом диаметр домена коллизий не превышает допустимый для данной конфигурации предел. Коммутатор (switching hub), входящий в состав сети, изображённой на рис. 9.28, считается оконечным устройством, так как коллизии через него не распространяются. Поэтому 2-километровый сегмент оптоволоконного кабеля, соединяющий этот коммутатор с маршрутизатором (router), не учитывается при расчёте диаметра домена коллизий сети Fast Ethernet. Сеть удовлетворяет правилам первой модели.

Проверим теперь её по второй модели. Наихудшие пути в домене коллизий: от DTE1 к DTE2 и от DTE1 к коммутатору (switching hub). Оба пути состоят из трёх сегментов на витой паре, соединённых двумя

повторителями класса II. Два сегмента имеют предельно допустимую длину 100 м. Длина сегмента, соединяющего повторители, равна 5 м.

Предположим, что все три рассматриваемых сегмента являются сегментами 100BASE-TX и в них используется витая пара категории 5. В табл. 9.3 приведены величины времени двойного оборота для рассматриваемых путей. Сложив числа из второго столбца этой таблицы, получим 511,96 би – это и будет время двойного оборота для наихудшего пути.

Таблица 9.3

Время двойного оборота сети рис. 9.28

Компонент пути	Время двойного оборота, би
Пара терминалов с интерфейсами TX	100
Сегмент на витой паре категории 5 (100 м)	111,2
Сегмент на витой паре категории 5 (100 м)	111,2
Сегмент на витой паре категории 5 (5 м)	5,56
Повторитель класса II	92
Повторитель класса II	92

Следует заметить, что в данном случае нет страхового запаса в 4 би, так как в этом примере используются наихудшие значения задержек, приведённые в табл. 9.2. Реальные временные характеристики компонентов Fast Ethernet могут отличаться в лучшую сторону.

9.2.3. Задание для выполнения

Требуется оценить работоспособность 100-мегабитной сети Fast Ethernet в соответствии с первой и второй моделями.

Конфигурации сети приведены в табл. 9.4. Топология сети представлена на рис. 9.29–9.30.

Таблица 9.4

Варианты заданий

No	Сегмент 1	Сегмент 2	Сегмент 3	Сегмент 4	Сегмент 5	Сегмент 6
1.	100BASE-TX, 100 м	100BASE-TX, 95 м	100BASE-TX, 80 м	100BASE-TX, 5 м	100BASE-TX, 100 м	100BASE-TX, 100 м
2.	100BASE-TX, 15 м	100BASE-TX, 5 м	100BASE-TX, 5 м	100BASE-FX, 400 м	100BASE-TX, 10 м	100BASE-TX, 4 м
3.	100BASE-TX, 60 м	100BASE-TX, 95 м	100BASE-TX, 10 м	100BASE-TX, 10 м	100BASE-TX, 90 м	100BASE-TX, 95 м

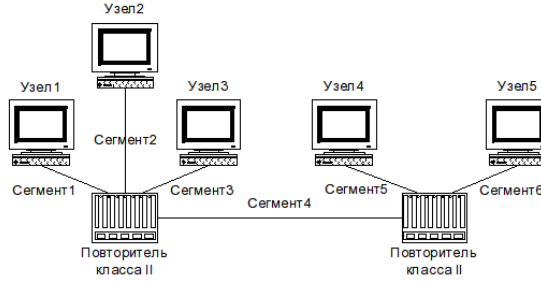


Рис. 9.29. Топология сети 1

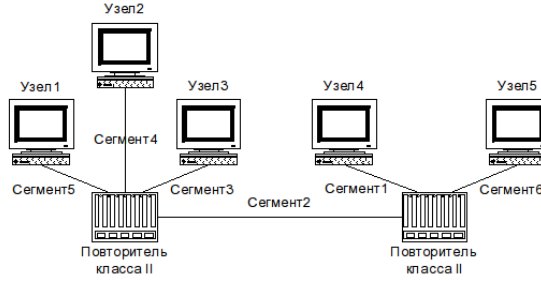


Рис. 9.30. Топология сети 2

9.3. Лабораторная работа. Знакомство с Packet Tracer. Моделирование простой сети

9.3.1. Цели работы

Изучение принципов построения сетей передачи данных и принципов настройки сетевого оборудования.

9.3.2. Предварительные сведения. Запуск, настройка, краткое описание интерфейса Packet Tracer

Packet Tracer — симулятор сети передачи данных, выпускаемый фирмой Cisco Systems.

С помощью данного симулятора можно строить модели сетей передачи данных, изучать настройки и принципы функционирования сетевого оборудования производителя, проводить диагностику работоспособности моделируемой сети.

Для запуска Packet Tracer, установленного под ОС Linux, достаточно в командной строке терминала ввести команду

```
packettracer &
```

Рабочее пространство Packet Tracer представлено на рис. 9.31.

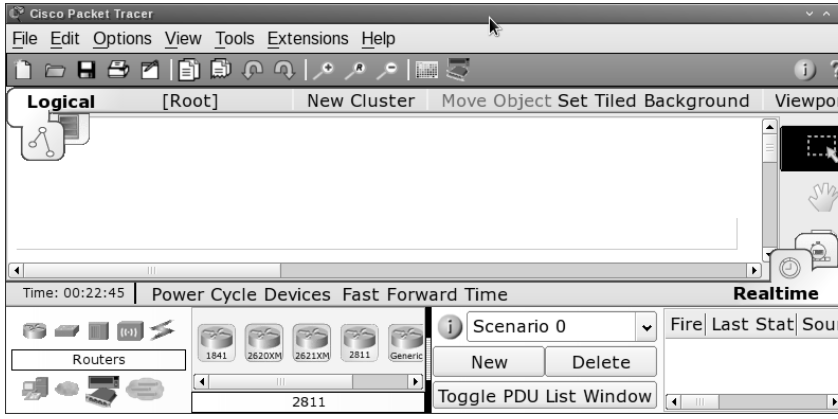


Рис. 9.31. Рабочее пространство Packet Tracer

По умолчанию интерфейс Packet Tracer — английский. Для изменения его на русский необходимо в меню выбрать «Options», «Preferences», вкладку «Interface» и указать язык «RUSSIANxx.ptl»,

где xx обозначает версию симулятора (рис. 9.32). Затем нужно нажать «Change Language» и перезапустить симулятор.

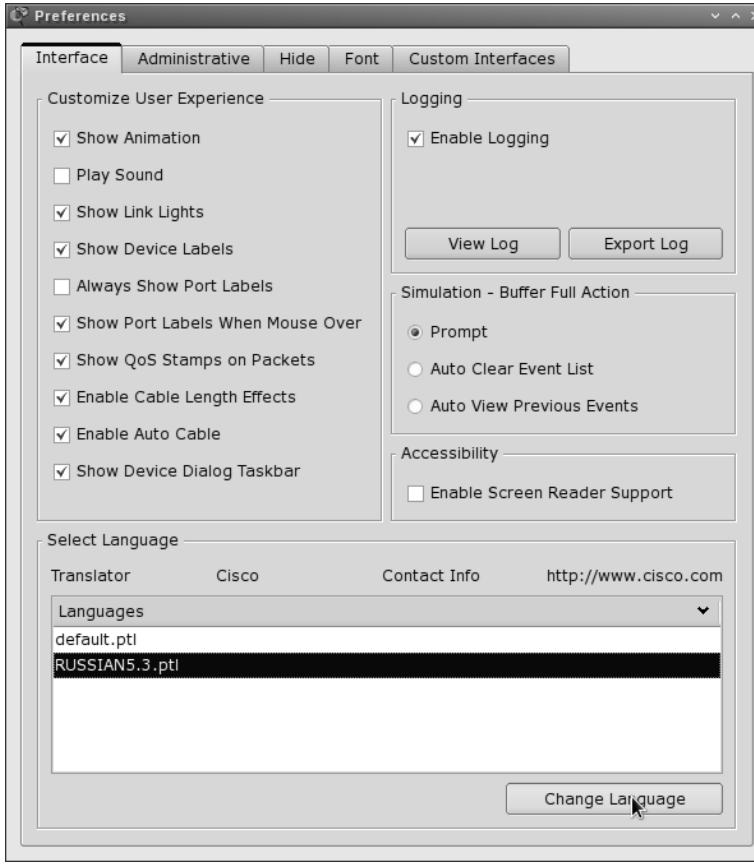


Рис. 9.32. Изменения языка интерфейса Packet Tracer

Основное окно программы (рис. 9.33) содержит программное меню (1), правое графическое окно с пиктограммами инструментов для работы с проектом и его объектами (2), меню выбора типа устройства (3), меню выбора устройства определённого типа (4).

Меню (1) позволяет создать, открыть, сохранить или распечатать проект, скопировать и вставить элемент, масштабировать рабочее пространство проекта.

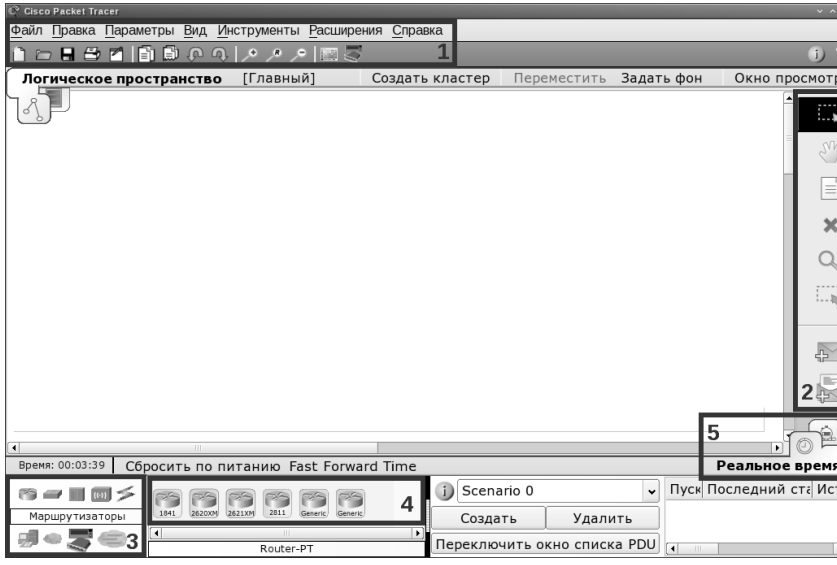


Рис. 9.33. Меню Packet Tracer

Меню (2) содержит инструменты выделения одного или нескольких объектов проекта, прокрутки проекта, добавления подписи к элементу проекта, удаления элемента проекта. Инструмент, напоминающий лупу, используется для просмотра содержимого таблиц ARP, NAT, таблицы маршрутизации и т.д. После этого инструмента расположены инструмент изменения размеров объекта, инструменты для эмулирования отправки с последующим отслеживанием произвольного пакета данных внутри проекта.

В меню (3) можно выбрать тип устройства: концентратор, коммутатор, маршрутизатор, тип соединения, оконечные и пользовательские устройства.

В меню (4) в зависимости от выбранного в меню (3) типа оборудования можно выбрать конкретное устройство.

Вкладка (5) позволяет переключаться с режима работы в реальном времени в режим симуляции и бывает полезна, если нужно более детально изучить, например, движение передаваемых от устройства к устройству данных, форматы конкретных пакетов.

9.3.3. Моделирование простейшей сети. Протокол ICMP

Постановка задачи

Требуется построить топологию сети из двух коммутаторов и четырёх пользовательских устройств (компьютеров), подсоединённых по два к каждому коммутатору. На пользовательских узлах нужно задать статическую адресацию из одного адресного пространства. Затем требуется изучить работу протокола ICMP.

Порядок выполнения работы

Создать новый проект (например, `lab03-01.pkt`).

В рабочем пространстве разместить 2 коммутатора, например, Cisco-2950, 4 оконечных устройства (рис. 9.34). Подключить оконечные устройства к коммутаторам, используя прямое соединение, и коммутаторы между собой, используя кроссовое соединение.

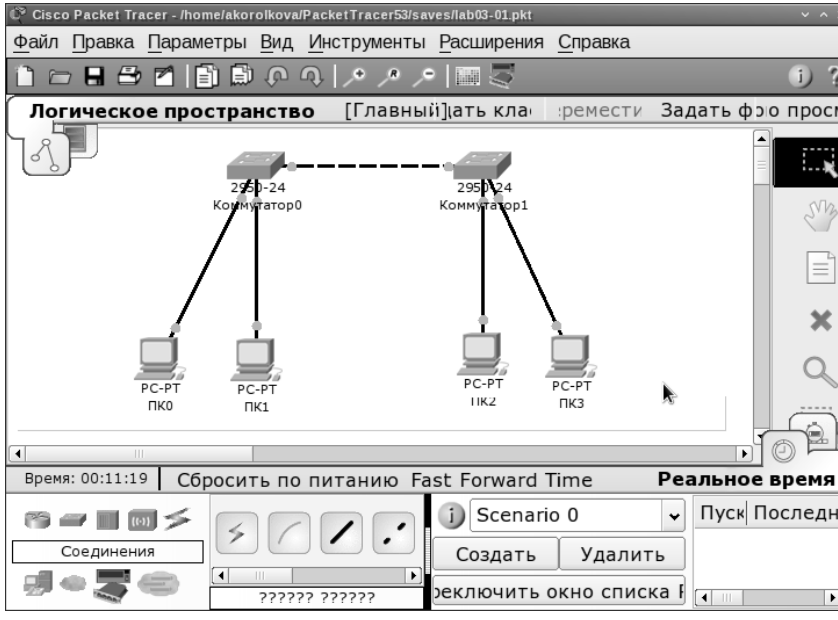


Рис. 9.34. Модель простой сети

Щёлкнув последовательно на каждом оконечном устройстве, задайте статический адрес из диапазона 192.168.1.2 – 192.168.1.5 (рис. 9.35).

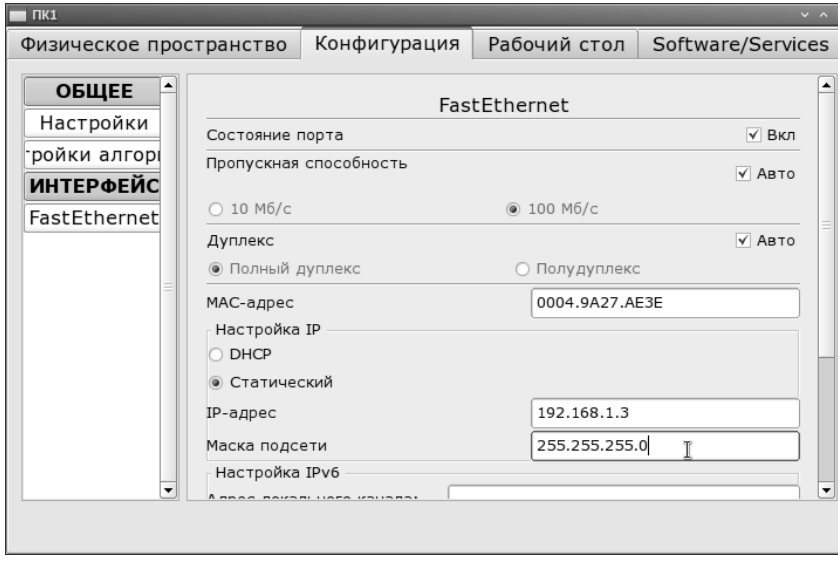


Рис. 9.35. Статическая адресация на оконечном устройстве

На одном из оконечных устройств запустите командную строку (рис. 9.36) и с помощью команды `ipconfig /all` посмотрите его сетевые настройки (рис. 9.37). Проверьте доступность другого узла сети с помощью команды `ping`.

В основном окне проекта перейдите из режима реального времени в режим симуляции. В командной строке одного из оконечных устройств повторите команду `ping`. С помощью кнопки «Захват / Вперёд» проследите движение пакета ICMP от одного оконечного устройства к другому (рис. 9.38).

Щёлкнув на значке пакета, откройте окно информации о PDU на устройстве и изучите его детали (рис. 9.39, 9.40). Используя кнопку «Проверь себя», ответьте на вопросы. Проделайте эту операцию на каждом этапе следования пакета и дайте пояснения об изменениях. Изучите информацию о PDU, передаваемых между коммутаторами (рис. 9.41, 9.42). Изучите изменения формата передаваемых пакетов других протоколов (например, telnet).

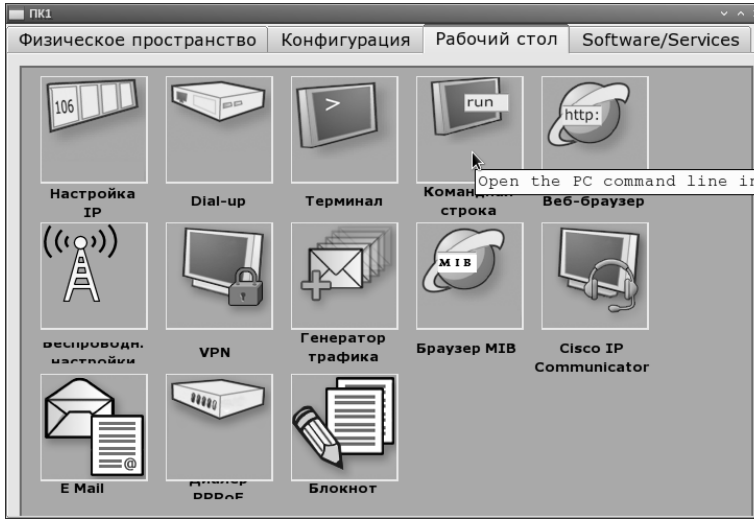


Рис. 9.36. Запуск командной строки на оконечном устройстве

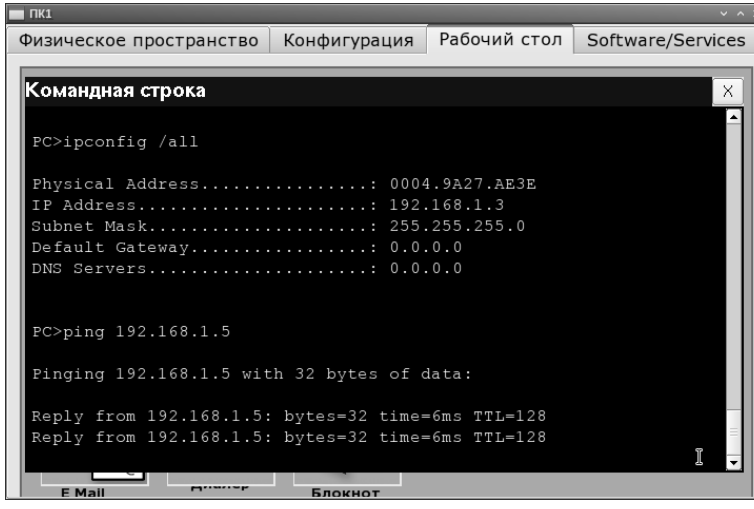


Рис. 9.37. Выполнение команды ipconfig на оконечном устройстве

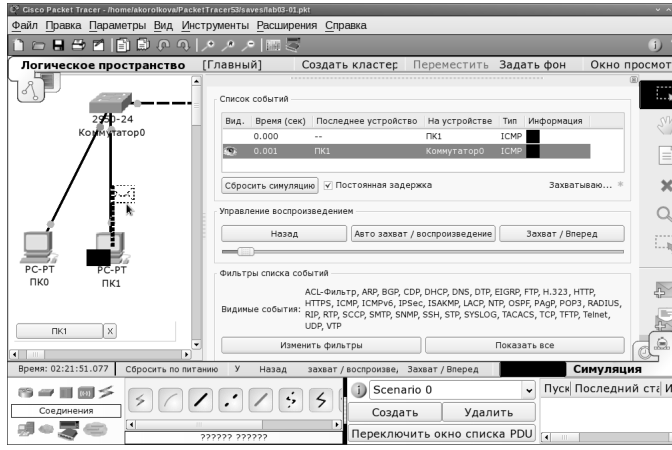


Рис. 9.38. Выполнение команды ping в режиме симуляции

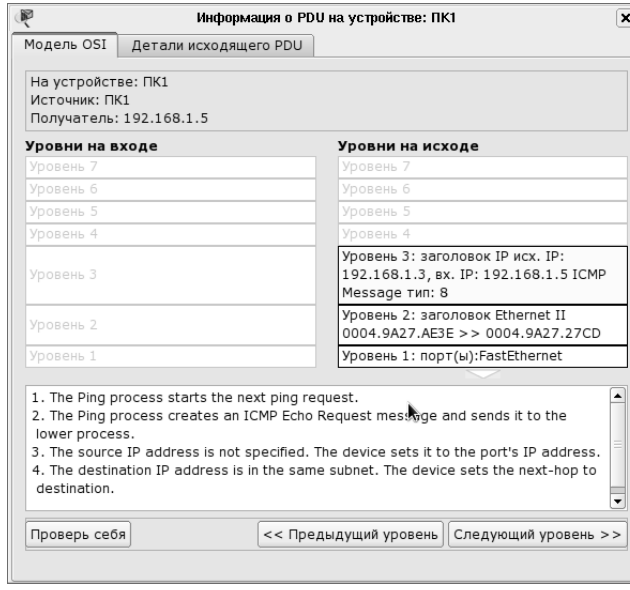


Рис. 9.39. Информация о PDU: уровень OSI

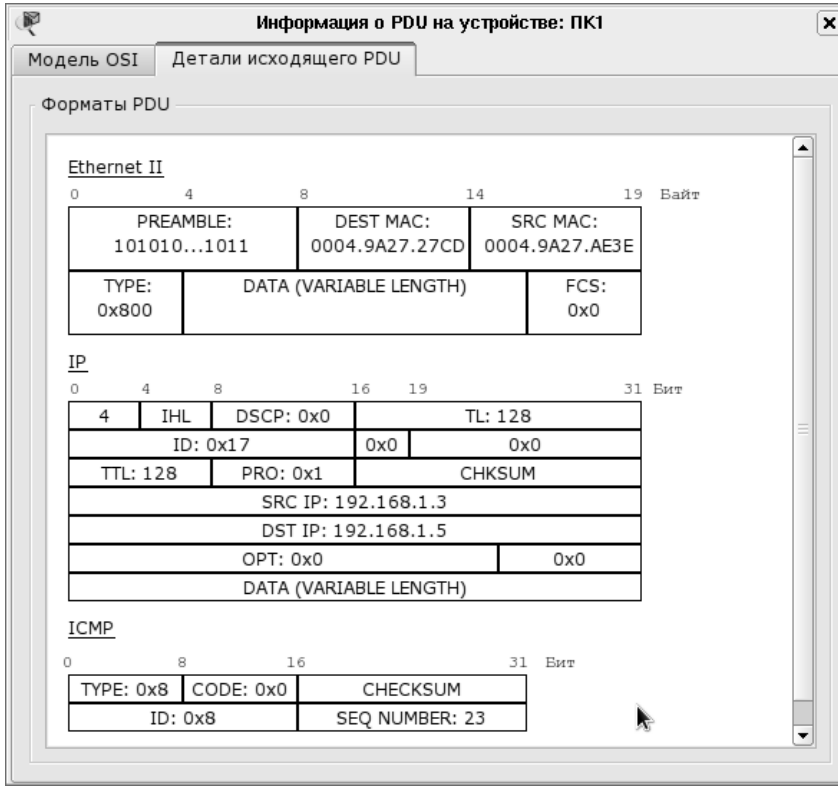


Рис. 9.40. Информация о PDU: форматы пакетов

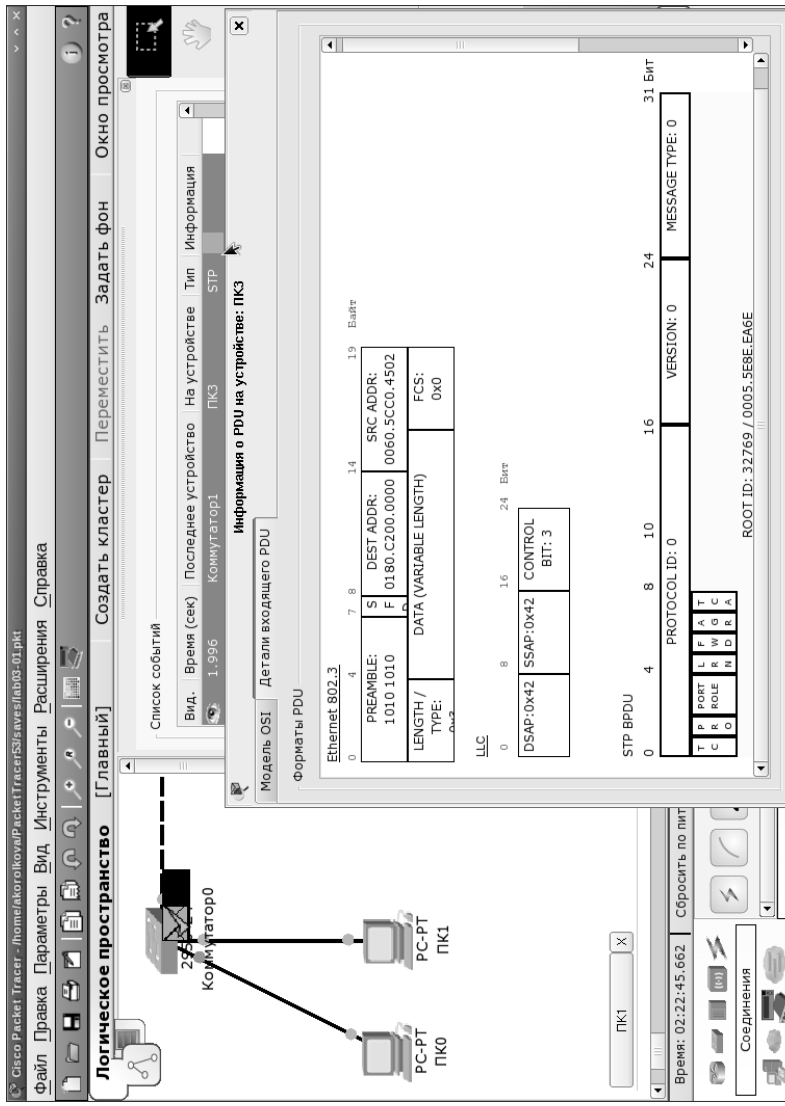


Рис. 9.41. Информация о PDU протокола STP

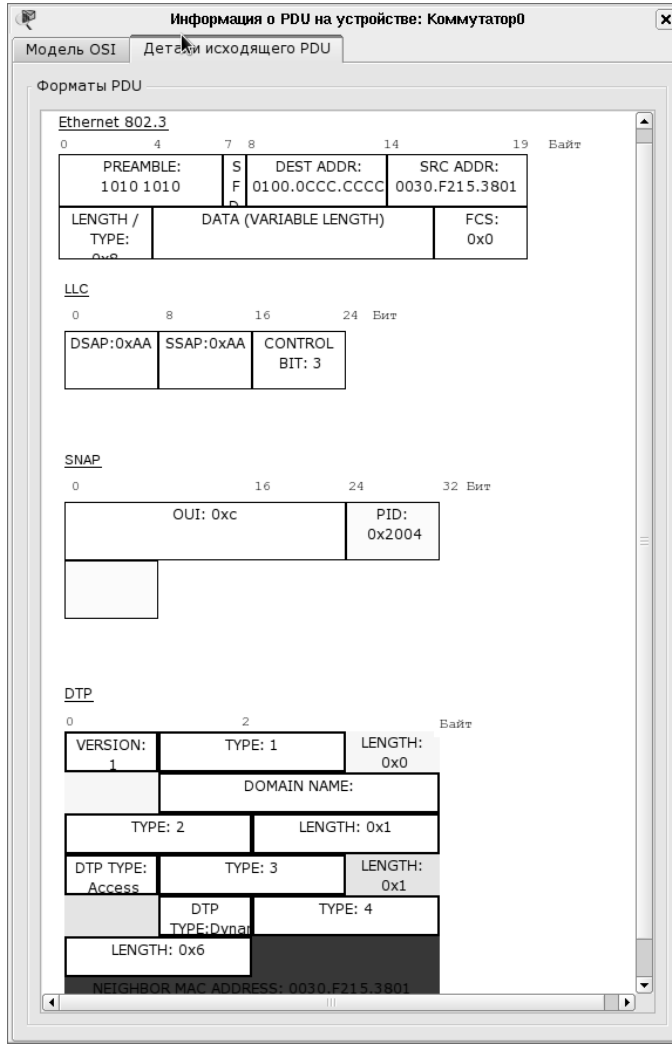


Рис. 9.42. Информация о PDU, передаваемых между коммутаторами

9.3.4. Конфигурирование оборудования Cisco

Для конфигурирования оборудования Cisco предусмотрен консольный кабель, на одном конце которого имеется разъём RJ45, подключаемый в соответствующий порт на оборудовании Cisco, а на другом — RS-232, подключаемый к COM-порту компьютера, с которого будет производится настройка. Для управления через COM-порт в Windows системах можно использовать HyperTerminal, а в Linux, например, minicom.

После подключения оборудования и включения питания на экране появляется приглашение пользовательского режима (отображается символом >), например

```
Router>
```

или

```
Switch>
```

В Packet Tracer соответственно, чтобы войти в режим конфигурирования оборудования, необходимо разместить это оборудование в рабочем пространстве, щёлкнуть на нём и перейти на вкладку CLI (рис. 9.43) командного интерфейса IOS (Internetwork Operating System).

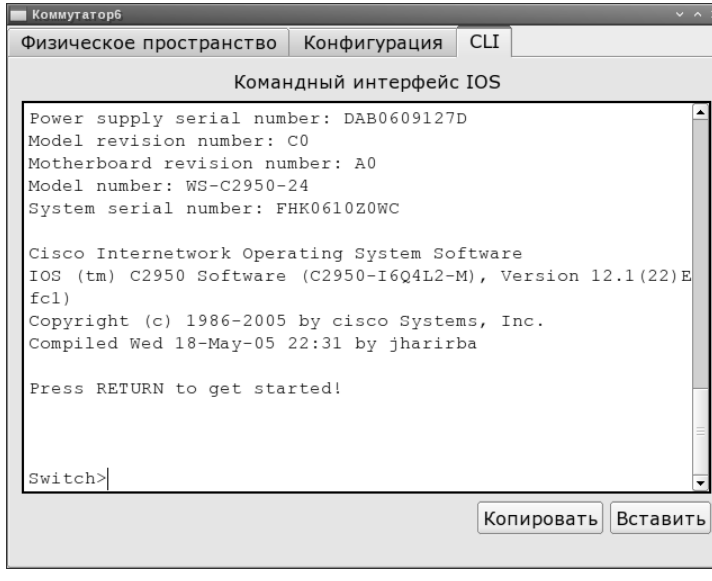


Рис. 9.43. Командный интерфейс IOS в Packet Tracer

Основные команды конфигурирования оборудования Cisco

Пользовательский режим на оборудовании Cisco предназначен только для просмотра конфигурации устройства и ввода простейших команд диагностики, например `ping`.

Для просмотра доступных команд используется команда `?` знака вопроса и нажатие клавиши Enter (рис. 9.44).

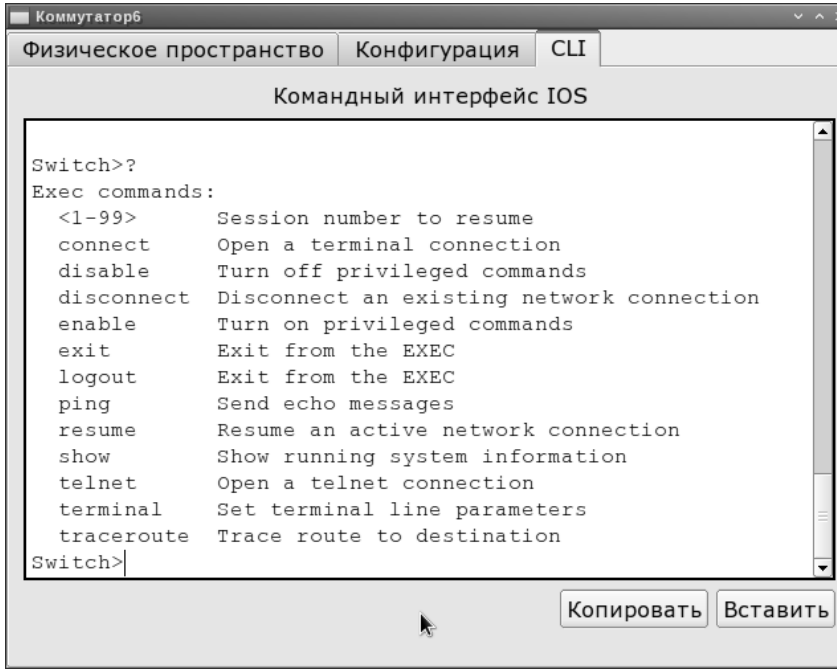


Рис. 9.44. Вызов списка команд, доступных в пользовательском режиме

Чтобы перейти в привилегированный режим и иметь возможность настраивать оборудование, вводится команда `enable` и приглашение заменяется на `#`, например:

```
Router#
```

или

```
Switch#
```

Список доступных в привилегированном режиме команд продемонстрирован на рис. 9.45 для коммутаторов и на рис. 9.46 для маршрутизаторов.

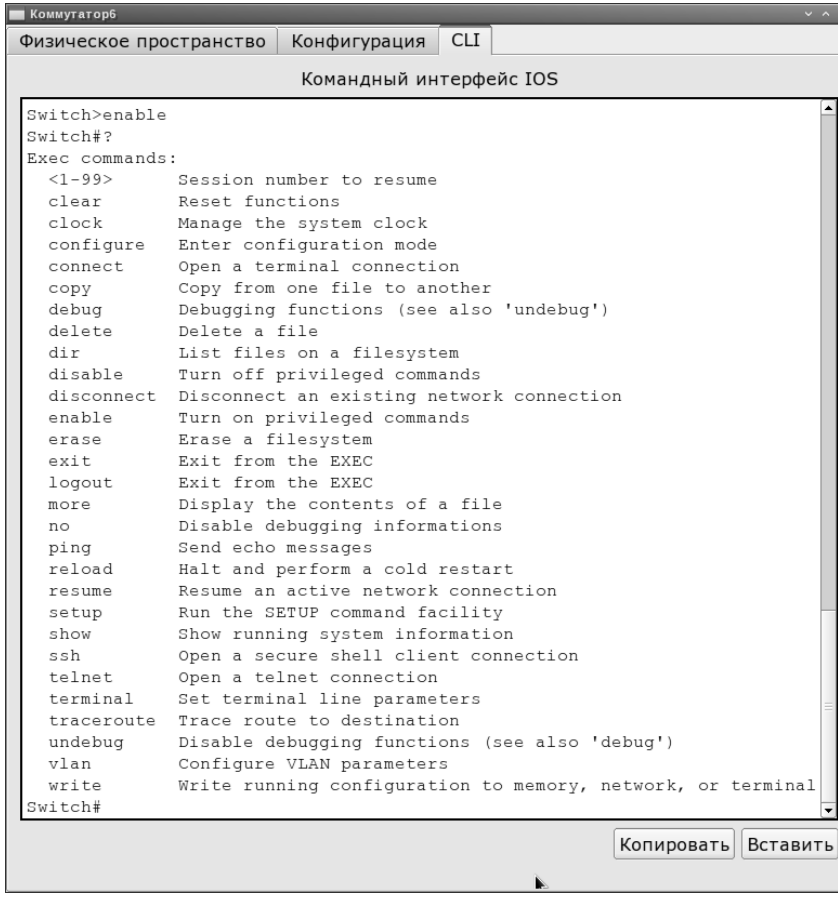


Рис. 9.45. Вызов списка команд коммутатора, доступных в привилегированном режиме

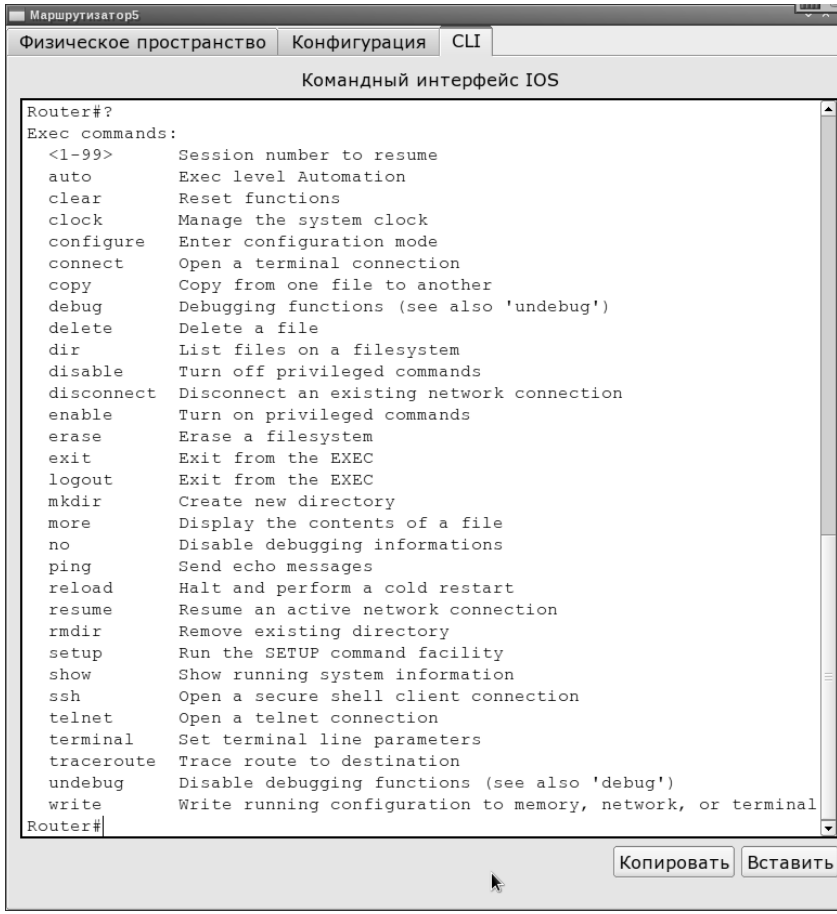


Рис. 9.46. Вызов списка команд маршрутизатора, доступных в привилегированном режиме

Для упрощения администрирования в устройствах Cisco применяется концепция VLAN-доменов.

VLAN (Virtual Local Area Network) — логическая («виртуальная») локальная компьютерная сеть, представляющая собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения.

VLAN можно назначить:

- определённому порту устройства;
- определённому MAC-адресу;
- определённому протоколу;
- по данным аутентификации пользователя или устройства.

По умолчанию на каждом порту устройства Cisco имеется сеть VLAN1 или VLAN управления. Сеть управления не может быть удалена, однако могут быть созданы дополнительные сети VLAN и этим альтернативным VLAN могут быть дополнительно назначены порты.

Операционная система IOS понимает как укороченное написание команд, так и их дописывание. Клавиша `Tab` автоматически дополняет команду, которую вы начали писать, что позволяет существенно уменьшить количество опечаток и повысить скорость работы. Клавиша `Tab` дополнит команду, а знак вопроса выведет на экран список дальнейших возможных действий, а также небольшое описание к ним.

Рассмотрим некоторые общие и часто используемые команды:

- `show` — просмотр параметров устройства:
  - `show running-config` — просмотр текущей конфигурации настраиваемого оборудования;
  - `show startup-config` — просмотр стартовой конфигурации;
  - `show mac-address-table` — просмотр таблицы MAC-адресов;
- `configure terminal` — переход в режим конфигурации и настройки:
  - `hostname имя` — задать имя устройства;
  - `service password-encryption` — включить режим хранения паролей в файле конфигурации устройства в зашифрованном виде;
  - `line console номер` — перейти в настройку консольного подключения:
    - `login` — разрешение проверки пароля;
    - `password пароль` — задать пароль;
  - `line vti 0 4` — настройка определённых виртуальных терминалов (в данном случае с 0 до 4 терминала):
    - `login` — разрешение проверки пароля;
    - `password пароль` — задать пароль;
  - `enable password пароль` — назначить пароль привилегированного уровня;
  - `service password-encryption` — зашифровать систему паролей;

- `interface тип/номер` — настройка определённого интерфейса (например, `interface fastEthernet0/1`):
  - `ip address ip-адрес маска` — задать ip-адрес и маску;
  - `no shutdown` — поднять интерфейс;
  - `description текст` — задать текстовое пояснение назначения интерфейса;
- `vlan номер` — настройка определённого vlan;
- `interface vlan номер` — настройка определённого vlan на интерфейсах (например, `interface vlan 1`):
  - `ip address адрес маска` — задать ip-адрес и маску на интерфейсе vlan;
  - `no shutdown` — поднять интерфейс vlan;
  - `description текст` — задать текстовое пояснение назначения интерфейса vlan;
- `exit` — выход из режима конфигурирования;
- `write memory` — запись внесённых изменений в память оборудования;
- `disable` — выход из привилегированного режима в пользовательский;
- `logout` — выйти из режима консоли.

После настройки коммутатора рекомендуется сохранять его текущую конфигурацию. Информация помещается в энергонезависимую память и хранится там столько, сколько нужно. При необходимости все настройки могут быть восстановлены или сброшены. Формат команды:

`copy running-config startup-config` — команда для сохранения конфигурации;

`copy startup-config running-config` — команда для загрузки конфигурации.

Если, например, на коммутаторе 2-го уровня требуется задать ip-адрес, по которому можно обратиться к нему, то это можно реализовать, задав адрес на VLAN1:

```
Switch> enable
Switch# configure terminal
Switch(config)# interface vlan 1
Switch(config-if)# ip address  ip-адрес маска
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# exit
Switch# write memory
```

Для проверки связи между устройствами сети можно использовать команду `ping`. Результаты выполнения команды ping представлены в табл. 9.5.

Таблица 9.5

Результаты выполнения команды ping

Символ	Значение
!	успешный приём эхо-ответа
.	превышено время ожидания
U	пункт назначения недосягаем
C	перегрузка сети
I	выполнение команды прервано администратором
?	неизвестный тип пакета
&	пакет превысил значение параметра времени жизни TTL пакета

Задание

Схема моделируемой сети представлена рис. 9.47.

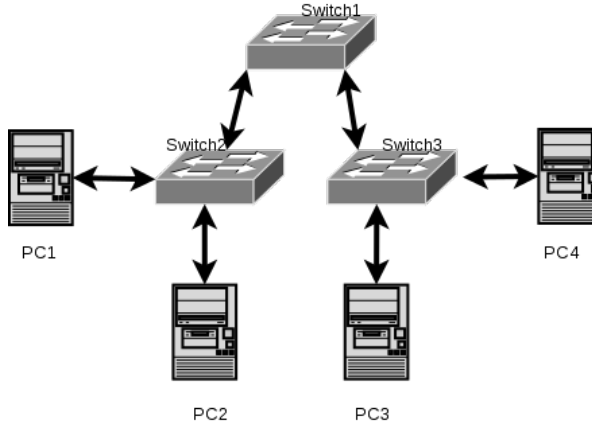


Рис. 9.47. Схема моделируемой сети

– Изменить имя коммутаторам Cisco.
– Обеспечить парольный доступ к привилегированному режиму на коммутаторах.

– Задать ip-адреса и маски коммутаторам (172.16.1.11/24, 172.16.1.12/24, 172.16.1.13/24).
– Задать ip-адреса и маски сетей персональным компьютерам (172.16.1.1/24, 172.16.1.2/24, 172.16.1.3/24, 172.16.1.4/24).
– Убедиться в достижимости всех объектов сети по протоколу IP.
– Переключившись в режим симуляции (его описание приведено в методических указаниях к предыдущей лабораторной работе) рассмотреть и пояснить процесс обмена данными по протоколу ICMP между устройствами (выполнив команду `ping` с одного компьютера на другой), пояснить роль протокола ARP в этом процессе. Детальное пояснение включить в отчёт.

Структура отчета по работе:

– Титульный лист.
– Задание.
– Схема сети.
– Ход работы: раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) и копий экранов (должна быть видна набранная команда и реакция системы, если она есть).
– Выводы.

9.4. Лабораторная работа. Packet Tracer. Настройка маршрутизаторов

9.4.1. Цели работы

Изучение принципов маршрутизации в IP-сетях и принципов настройки сетевого оборудования.

9.4.2. Предварительные сведения

Понятия DTE и DCE

Оконечное оборудование данных (Data Terminal Equipment, DTE) — термин для обозначения устройства, обеспечивающего интерфейс с пользователем. Обычно таким оборудованием является терминал или ЭВМ. На DTE исполняются пользовательские прикладные программы.

Оконечное оборудование канала данных (Data Circuit-terminating Equipment, DCE) обеспечивает подключение DTE к связному каналу для передачи преобразования и усиления сигнала, сгенерированного DTE.

Маршрутизатор с последовательным интерфейсом можно считать как устройством DCE (маршрутизатор только транслирует данные на сетевом уровне), так и устройством DTE (цепочки бит на выходе

маршрутизатор сам генерирует и сам обрабатывает принятые данные). Соответственно, корректнее говорить о типе портов (DTE/DCE), при этом имея в виду следующие соглашения:

- для порта DTE сигнал, обозначаемый как TxD (данные передатчика), является выходным, сигнал RxD (данные приемника) — входным;
- для порта DCE сигнал, обозначаемый как TxD, является входным (устройство должно передать данные в канал связи), а сигнал RxD — выходным (данные из канала, которые должны попасть на вход приемника DTE).

Из соглашений следует, что при соединении порта DTE с портом DCE одноименные сигнальные цепи должны соединяться через последовательное соединение точка-точка. При соединении однотипных портов (DTE-DTE или DCE-DCE) сигнальные цепи должны соединяться перекрестно (TxD-RxD, RxD-TxD), а управляющие — в соответствии с логикой протокола.

Для асинхронного режима передачи посимвольная синхронизация осуществляется старт-битами и внешняя подача синхронизации не требуется (скорости и допуски отклонения согласованы, и любое устройство имеет свой внутренний генератор). В синхронном режиме все последовательные интерфейсы используют внешние сигналы синхронизации, причем раздельные для передаваемых ST (Send Timing, Send Clock) и принимаемых RT (Receive Timing, Receive Clock) данных.

Для устройств DTE сигналы синхронизации, по которым работают сдвигающие регистры их приемников и передатчиков, являются входными. Источником синхронизации (сигнал RT) для принимаемых данных (RxD) практически однозначно является DCE (синхросигнал обычно выделяется из самосинхронизирующего сигнала линии связи).

Если синхронный канал сам навязывает свою синхронизацию, то для DTE первичным источником синхронизации будет DCE. DCE может потребовать и внешней синхронизации. Для этих целей порт DTE снабжают дополнительным выходом TT (Terminal Timing), от которого может синхронизироваться DCE. Но сам передатчик DTE будет синхронизироваться от входа ST, на который сигнал поступит либо от DCE, либо по перемычке в разъеме. Возможен вариант конфигурирования, когда сигнал TT заводится на передатчик внутри порта (не с разъема). В высокоскоростном интерфейсе для синхронных каналов (HSSI) источником синхронизации является только DCE, а сигнал TT является буферированным принятым сигналом ST. Такой «разворот» синхросигнала позволяет учесть задержки распространения сигнала в интерфейсном кабеле (сигналы TxD и TT идут параллельными путями, и фазовый перекос между ними будет небольшим). Для компенсации перекоса при высоких скоростях может применяться и инверсия сигнала синхронизации, которую включают при настройке

порта.

В названиях сигналов синхронизации передатчика бывает путаница, и внешним (external) сигналом синхронизации называют как сигнал от DTE, так и сигнал, по которому DTE в действительности синхронизирует свой приемник. Понять, что есть что, позволяет указание на источник (DTE или DCE), т.е., если источником является DCE, то DTE синхронизирует свой приёмник по сигналу от DCE, который и является в этом случае внешним.

Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) позволяет устройствам обмениваться основной конфигурационной информацией. CDP работает на втором (канальном) уровне модели OSI, поэтому он не является маршрутизируемым протоколом и работает только с непосредственно подключенными устройствами. Протокол CDP связывает физическую среду передачи данных более низкого уровня с протоколами более высокого сетевого уровня. Поэтому устройства, поддерживающие разные протоколы третьего уровня, могут узнавать друг друга.

При запуске устройства протокол CDP запускается автоматически, после чего он может автоматически определить соседние устройства, на которых также работает протокол CDP, и выдать информацию, например, об идентификаторах устройств и портов, аппаратной платформе.

Команда traceroute

Команды `traceroute` показывает адреса промежуточных интерфейсов (хопов) на пути пакетов в пункт назначения.

Для определения промежуточных маршрутизаторов `traceroute` отправляет узлу назначения серию ICMP-пакетов (по умолчанию 3 пакета), с каждым шагом увеличивая значение поля TTL («время жизни») на 1. Это поле обычно указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно ICMP-сообщение «time exceeded in transit», указывающее на невозможность доставки данных. Traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем `traceroute` повторяет отправку серии пакетов, но уже с TTL, равным 2, что заставляет первый маршрутизатор уменьшить TTL пакетов на единицу и направить их ко второму маршрутизатору. Второй маршрутизатор, получив пакеты с TTL=1, также возвращает «time exceeded in transit». Процесс повторяется до тех пор, пока пакет

не достигнет узла назначения. При получении ответа от этого узла процесс трассировки считается завершённым.

На оконечном хосте IP-дейтаграмма с TTL = 1 не отбрасывается и не вызывает ICMP-сообщения типа «срок истёк», а должна быть отдана приложению. Достижение пункта назначения определяется следующим образом: отсылаемые `traceroute` дейтаграммы содержат UDP-пакет с заведомо неиспользуемым номером порта на адресуемом хосте. Номер порта будет равен 33434 + (максимальное количество транзитных участков до узла) — 1. В пункте назначения UDP-модуль, получая подобные дейтаграммы, возвращает ICMP-сообщения об ошибке «порт недоступен». Таким образом, чтобы узнать о завершении работы, программе `traceroute` достаточно обнаружить, что поступило ICMP-сообщение об ошибке этого типа.

Address Resolution Protocol

Address Resolution Protocol (ARP) — протокол сетевого уровня, предназначенный для определения MAC-адреса по известному IP-адресу.

Типы сообщений ARP: запрос ARP (ARP request) и ответ ARP (ARP reply). Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP.

Отправитель определяет IP-адрес приёмника, просматривает свою ARP таблицу и определяет MAC-адрес приёмника. Если MAC- и IP-адреса приёмника присутствуют в ARP-таблице отправителя, то между ними устанавливается соответствие и приёмник использует его в ходе инкапсуляции IP-пакетов во фреймы канального уровня. MAC-адреса фреймов канального уровня берутся из ARP-таблиц. После этого фрейм по физическому каналу отправляется от отправителя к адресату.

Если записи в кэше ARP нет, то выполняется широковещательный запрос ARP. Запрос принимают все сетевые устройства в сегменте сети, но только устройство, имеющее запрашиваемый IP-адрес, реагирует на него, посылая отправителю информацию о MAC-адресе своего сетевого интерфейса со своим IP-адресом. Отправитель записывает MAC-адрес и IP-адрес в свою ARP-таблицу.

9.4.3. Моделирование сети со статической маршрутизацией

Постановка задачи

1. Построить топологию сети из трёх маршрутизаторов (рис. 9.48). Соединение между первым и вторым маршрутизаторами — Ethernet,

соединение между первым и третьим маршрутизаторами — последовательное соединение точка-точка (serial cable), причём первый маршрутизатор должен выполнять функции DCE-устройства, т.е. задавать синхронизацию, с тактовой частотой 64 Кбит/с, а третий маршрутизатор — DTE.

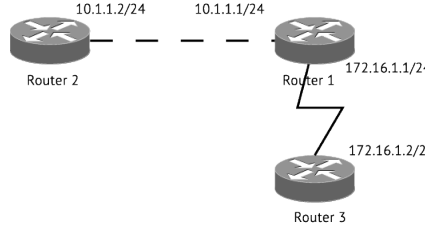


Рис. 9.48. Модель сети

2. Задать имена маршрутизаторам и IP-адреса их интерфейсам. Первый маршрутизатор (Router1) должен иметь 2 адреса: 10.1.1.1/24, 172.16.10.1/24. Второй маршрутизатор (Router2) должен иметь адрес 10.1.1.2/24, третий (Router3) — 172.16.10.2/24 (см. рис. 9.48).
3. Изучите состояния всех интерфейсов, на которых работает CDP.
4. На Router1 настройте возможность работы по протоколу telnet. С Router2 зайдите по на Router1 по telnet. Выведите информацию о подключённых на Router1 пользователях. На Router2 выведите информацию о запущенных сессиях, возобновите telnet-сессию, а затем закройте её.
5. Настройте сначала статическую маршрутизацию, а затем статическую маршрутизацию по умолчанию с Router2 на Router3 и с Router3 на Router2.
6. Оформите отчёт, зафиксировав в нём производимые вами действия.

Порядок выполнения работы

Создайте новый проект (например, `lab04-01.pkt`).

В рабочем пространстве разместите 3 маршрутизатора типа Generic и соедините их согласно требованиям задания (рис. 9.49).

Переименуйте маршрутизаторы, назвав их Router1, Router2, Router3. Для этого при конфигурации каждого маршрутизатора используйте следующую последовательность команд (например, для маршрутизатора 0):

```
Continue with configuration dialog? [yes/no]: no
Router>enable
Router#configure terminal
```

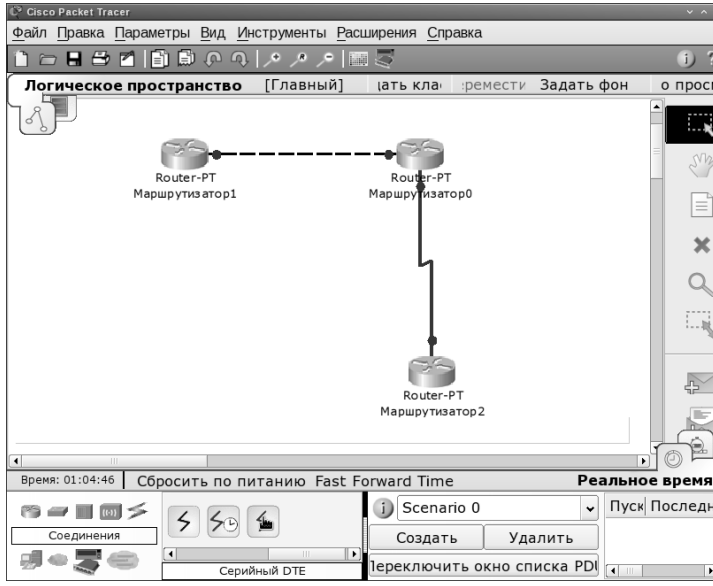


Рис. 9.49. Модель сети: 3 маршрутизатора типа Generic

```
Router(config)#hostname Router1
Router1(config)#exit
Router1#write memory
```

Соответственно, во вкладке «Конфигурация» замените «Отображаемое имя» (рис. 9.50).

С помощью значка боковой панели «Сделать пометку» обозначьте планируемое распределение адресного пространства сети. В результате получите следующую топологию сети (рис. 9.51).

Далее необходимо в соответствии с заданием настроить интерфейсы Ethernet маршрутизаторов, подняв их и задав им описание. Например, для Router1:

```
Router1#configure terminal
Router1(config)#interface f0/0
Router1(config-if)#no shutdown
Router1(config-if)#description Ethernet on Router1
Router1(config-if)#exit
Router1#write memory
Router1#show running-config
```

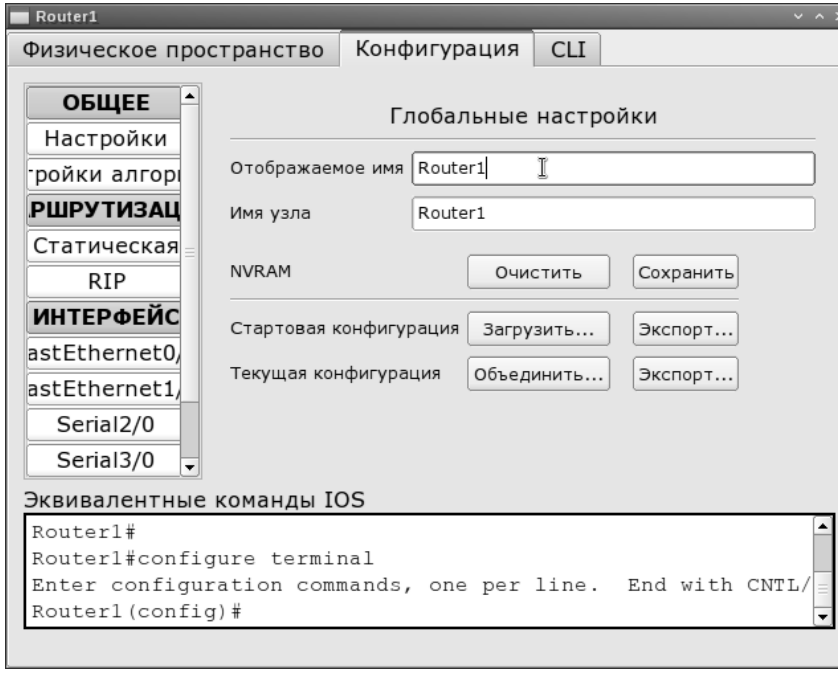


Рис. 9.50. Замена отображаемого имени маршрутизатора на вкладке «Конфигурация»

По аналогии настройте интерфейс на Router2.

Далее необходимо в соответствии с заданием настроить интерфейсы последовательного соединения маршрутизаторов, подняв их и задав им описание и тактовую частоту на DCE в 64 Кбит/с. Например, для Router1 (по заданию является DCE):

```
Router1#show controllers s2/0
Router1#configure terminal
Router1(config)#interface s2/0
Router1(config-if)#clock rate 64000
Router1(config-if)#no shutdown
Router1(config-if)#description Serial on Router1
Router1(config-if)#exit
Router1(config)#exit
Router1#write memory
Router1#show running-config
```

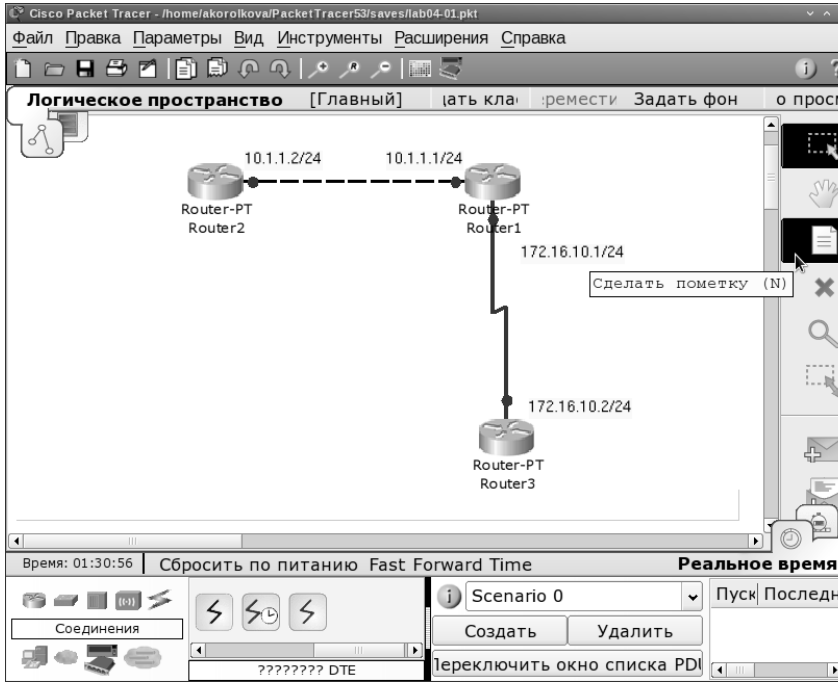


Рис. 9.51. Модель сети с переименованными маршрутизаторами

По аналогии настройте интерфейс на Router3, учитывая, что он является DTE (тактовую частоту задавать не требуется).

На Router1 введите команду для вывода состояния всех интерфейсов, на которых работает CDP:

```
Router1#show cdp interface
```

и убедитесь, что интерфейсы f0/0 и s2/0 подняты и посылают CDP-пакеты.

Затем получите краткую и полную информацию о непосредственно подключённых к Router1 устройствах:

```
Router1#show cdp neighbors
Router1#show cdp neighbors detail
```

Получите информацию об устройстве Router3:

```
Router1#show cdp entry Router3
```

Задайте ip-адреса маршрутизаторам и с помощью команд `ping` и `traceroute` проверьте доступность всех маршрутизаторов. Убедитесь, что с Router3 команды покажут недоступность адресов 10.1.1.1 и 10.1.1.2.

На Router1 настройте возможность работы по протоколу telnet. Необходимо, чтобы сетевое устройство принимало telnet-сессии и было защищено паролем. Каждая так называемая линия (line vti) в сетевом устройстве потенциально представляет активную telnet-сессию, которую устройство может поддерживать. Наши сетевые устройства поддерживают до 5 линий, назначенные на виртуальные терминалы vty:

```
Router1#configure terminal
Router1(config)#line vty 0 4
Router1(config-line)#password Cisco
Router1(config-line)#login
Router1(config-line)#exit
Router1(config)#service password-encryption
Router1(config)#exit
Router1#write memory
```

С Router2 зайдите на Router1 по telnet:

```
Router2>telnet 10.1.1.1
```

Затем выведите информацию о подключённых на Router1 пользователях, введя

```
Router1#show users
```

Нажмите одновременно клавиши control-shift-6, потом отпустите и сразу нажмите клавишу `x`. Имя сетевого устройства должно поменяться на Router2. Выведите информацию о запущенных сессиях:

```
Router2>show sessions
```

Вернитесь в telnet-сессию на Router1, используя клавиши control-shift-6 и `1` вместо `x`. Посмотрите конфигурацию Router1.

Вернитесь на Router2 и возобновите telnet-сессию, а затем закройте её:

```
Router2>resume 1
  [Resuming connection 1 to 10.1.1.1 ... ]

Router1>
Router2>disconnect 1
```

Настройте статическую маршрутизацию.

Для просмотра таблицы маршрутов следует использовать команду `show ip route`. Маршрут на непосредственно подсоединённые сети отображается на интерфейс маршрутизатора, к которому они присоединены. Таблица маршрутов отображает сетевые префиксы (адреса сетей) на выходные интерфейсы.

Для направления пакетов к другим адресатам необходимо в таблицу маршрутизации включить дополнительные маршруты (статически или динамически). Статические маршруты не меняются самим маршрутизатором. Динамические маршруты изменяются самим маршрутизатором автоматически при получении информации о смене маршрутов от соседних маршрутизаторов.

Для конфигурации статической маршрутизации в маршрутизаторах Cisco используют две версии команды `ip route`:

1. `ip route АдресСетиНазначения МаскаСетиНазначения Интерфейс`
   Команда указывает маршрутизатору, что все пакеты, предназначенные для *АдресСетиНазначения-МаскаСетиНазначения*, следует направлять на свой интерфейс *Интерфейс*. Если интерфейс *Интерфейс* имеет тип Ethernet, то MAC-адреса исходящих пакетов будут широковещательными.
2. `ip route АдресСетиНазначения МаскаСетиНазначения Адрес`
   Команда указывает маршрутизатору, что все пакеты, предназначенные для *АдресСетиНазначения-МаскаСетиНазначения*, следует направлять на тот свой интерфейс, из которого достижим IP-адрес *Адрес*. Как правило, *Адрес* — это адрес следующего хопа по пути к *АдресСетиНазначения*. Выходной интерфейс и физические адреса исходящих пакетов определяются маршрутизатором по своим ARP-таблицам на основании IP-адреса *Адрес*.

Для сетей типа Ethernet рекомендуется всегда использовать форму (2) команды `ip route`. Ethernet-интерфейс на маршрутизаторе, как правило, соединён с несколькими Ethernet-интерфейсами других устройств в сети. Указание в команде `ip route` IP-адреса позволит маршрутизатору правильно сформировать физический адрес выходного пакета по своим ARP-таблицам.

Подсоединитесь к маршрутизатору Router2 и посмотрите таблицу маршрутов. Добавьте маршрут к сети 172.16.10.0/24 через адрес 10.1.1.1 ближайшего хопа на пути к этой сети:

```
Router2#show ip route
Router2#configure terminal
Router2(config)#ip route 172.16.10.0 255.255.255.0 10.1.1.1
Router2(config)#exit
Router2#write memory
Router2#show ip route
Router2#ping 172.16.10.1
```

Подсоединитесь к маршрутизатору Router3 и посмотрите таблицу маршрутов. Добавьте маршрут к сети 10.1.1.0/24 через адрес 172.16.10.1 ближайшего хопа на пути к этой сети:

```
Router3#show ip route
Router3(config)#ip route 10.1.1.0 255.255.255.0 172.16.10.1
Router3(config)#exit
Router3#write memory
Router3#show ip route
Router3#ping 10.1.1.2
```

Сетевые устройства Router2 и Router3 имеют только по одному выходу во внешний мир: через интерфейсы с адресами 10.1.1.1 и 172.16.10.1 соответственно. Поэтому можно не определять? на какие подсети маршрутизируются пакеты, и использовать так называемую *маршрутизацию по умолчанию*.

Маршруты по умолчанию используются, когда маршрутизатор не может поставить в соответствие сети назначения строку в таблице маршрутов. В этом случае маршрутизатор должен использовать маршрут по умолчанию для отсылки пакетов другому маршрутизатору. Следующий маршрутизатор будет иметь маршрут к этой сети назначения или иметь свой маршрут по умолчанию к третьему маршрутизатору и т.д. В конечном счёте пакет будет маршрутизирован на маршрутизатор, имеющий маршрут к сети назначения.

Маршрут по умолчанию может быть статически введен администратором или динамически получен из протокола маршрутизации. Так как все IP-адреса принадлежат сети 0.0.0.0 с маской 0.0.0.0, то в простейшем случае надо использовать команду

```
ip route 0.0.0.0 0.0.0.0 адрес_следующего_хопа
```

или

```
ip route 0.0.0.0 0.0.0.0 выходной_интерфейс
```

Ручное задание маршрута по умолчанию на каждом маршрутизаторе подходит для простых сетей. В сложных сетях необходимо организовать динамический обмен маршрутами по умолчанию.

Сначала удалите старые маршруты на Router2:

```
Router2#configure terminal
Router2(config)#no ip route 172.16.10.0 255.255.255.0 10.1.1.1
```

и на Router3:

```
Router3#configure terminal
Router3(config)#no ip route 10.1.1.0 255.255.255.0 172.16.10.1
```

Назначьте маршруты по умолчанию:

```
Router2#configure terminal
Router2(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1
Router2(config)#exit
```

```
Router2#write memory
Router2#show ip route

Router3#configure terminal
Router3(config)#ip route 0.0.0.0 0.0.0.0 172.16.10.1
Router3#write memory
Router3#show ip route
Router3#ping 10.1.1.2
```

Проверьте доступность всех узлов сети.

Сохраните своей проект и оформите первую часть отчёта, зафиксировав в нём производимые вами действия.

9.4.4. Задание для самостоятельной работы

1. Создайте новый проект (например, `lab04-02.pkt`). Постройте в Packet Tracer топологию, представленную на рис. 9.52.

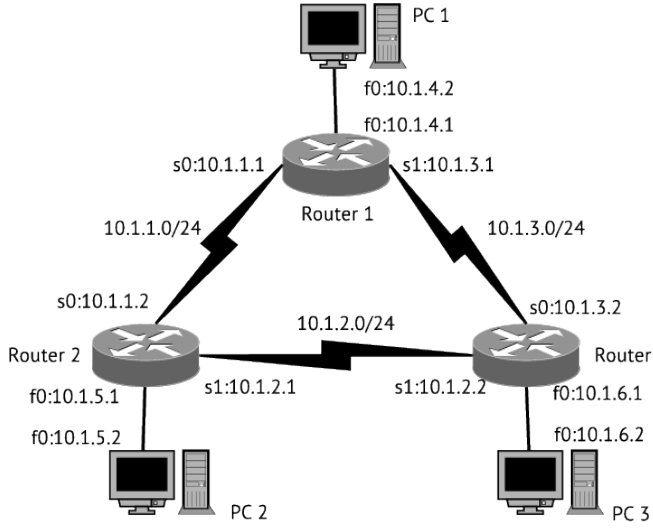


Рис. 9.52. Модель сети для самостоятельной работы

2. На каждом маршрутизаторе настройте используемые интерфейсы и определите соседей.
3. Назначьте интерфейсам сети адреса согласно рис. 9.52.
4. Назначьте шлюзы по умолчанию для компьютеров.

5. Проверьте факт назначения адресов (на каждом маршрутизаторе выполните `show running-config` и `show ip interface brief`). Для компьютеров используйте команду `ipconfig`.
6. Проверьте правильность назначения адресов путём выполнения на каждом маршрутизаторе команды `ping` к непосредственным соседям.
7. Настройте на маршрутизаторах статическую маршрутизацию. На каждом маршрутизаторе посмотрите таблицу маршрутизации. Сохраните конфигурацию.
8. На каждом компьютере выполните команду трассировки `tracert` других компьютеров.
9. Сохраните проект. Подготовьте отчёт, зафиксировав в нём производимые вами действия.

9.4.5. Моделирование сети с динамической маршрутизацией

Предварительные сведения

Динамическая маршрутизация может быть реализована с помощью одного или нескольких протоколов. Протоколы для работы внутри автономных систем называют *внутренними протоколами шлюзов (Interior Gateway Protocols, IGP)*, а протоколы для работы между автономными системами называют *внешними протоколами шлюзов (Exterior Gateway Protocols, EGP)*. К протоколам IGP относятся RIP, RIPv2, IGRP, EIGRP, OSPF и IS-IS. Протоколы EGP3 и BGP4 относятся к EGP.

Маршрутизаторы используют метрики для оценки или измерения маршрутов. Например, протокол RIP использует в качестве метрики количество переходов (хопов), а EIGRP — сложную комбинацию факторов, включающую полосу пропускания канала и его надёжность.

Результаты работы маршрутизирующих протоколов заносятся в таблицу маршрутов, которая постоянно изменяется при смене ситуации в сети. Простые протоколы RIP и IGRP распространяют информацию о таблицах маршрутов через все интерфейсы маршрутизатора в широковещательном режиме без уточнения точного адреса конкретного соседнего маршрутизатора. Соседний маршрутизатор, получая широковещательное сообщение, сравнивает информацию со своей текущей таблицей маршрутов, добавляет в неё маршруты к новым сетям или маршруты к известным сетям с лучшей метрикой, удаляет несуществующие маршруты, добавляет свои собственные значения к метрикам полученных маршрутов. Новая таблица маршрутизации снова распространяется по соседним маршрутизаторам.

Если маршрутизатор работает с протоколом OSPF, то он строит полную базу данных всех состояний связи в своей области. Каждый маршрутизатор затем самостоятельно выполняет *алгоритм поиска*

наикратчайшего пути (Shortest Path First, SPF) на своём собственном отображении сети или базе данных состояний связи для определения лучшего пути, который заносится в таблицу маршрутов. Эти пути к другим сетям формируют дерево с вершиной в виде локального маршрутизатора.

Маршрутизаторы извещают о состоянии своих связей все маршрутизаторы в области. Такое извещение называют *LSA (Link-State Advertisements)*. Имеет место начальный наплыв LSA пакетов для построения базы данных состояний связи. Обновление маршрутов происходит только при изменении состояний связи или в течение определённого интервала времени. Если состояние связи изменилось, то частичное обновление пересылается немедленно. Оно содержит только состояния связей, которые изменились, а не всю таблицу маршрутов.

Порядок выполнения работы

Загрузите проект лабораторной работы раздела 9.4.3 (`lab04-01.pkt`), пересохраните его под другим именем (например, `lab04-03.pkt`).

Посмотрите таблицы маршрутизации на Router2 и Router3 и отключите статическую маршрутизацию по умолчанию:

```
Router2#enable
Router2#show ip route
Router2#ping 172.16.10.2
Router2#configure terminal
Router2(config)#no ip route 0.0.0.0 0.0.0.0 10.1.1.1
Router2(config)#exit
Router2#ping 172.16.10.2
Router2#write memory

Router3#enable
Router3#show ip route
Router3#configure terminal
Router3(config)#no ip route 0.0.0.0 0.0.0.0 172.16.10.1
Router3(config)#exit
Router3#write memory
```

Настройка RIP на маршрутизаторах

Включите RIP на всех маршрутизаторах и посмотрите изменение в их конфигурации:

```
Router1#configure terminal
Router1(config)#router rip
Router1(config-router)#network 172.16.10.0
```

```
Router1(config-router)#network 10.1.1.0
Router1(config-router)#exit
Router1(config)#exit
Router1#write memory
Router1#show running-config

Router2#configure terminal
Router2(config)#router rip
Router2(config-router)#network 10.1.1.0
Router2(config-router)#exit
Router2(config)#exit
Router2#write memory
Router2#show running-config

Router3#configure terminal
Router3(config)#router rip
Router3(config-router)#network 172.16.10.0
Router3(config-router)#exit
Router3(config)#exit
Router3#write memory
Router3#show running-config
```

Сеть 10.1.1.0/24 должна быть воспринята маршрутизаторами как сеть 10.0.0.0/8, а сеть 172.16.10.0/24 — как сеть 172.16.0.0/16. Это связано с классами IP-адресов.

С помощью команды `show ip route` определите параметры, по которым работает протокол RIP. С помощью команды `show ip route` посмотрите таблицы маршрутов. С помощью команд `ping` и `traceroute` проверьте доступность других маршрутизаторов.

С помощью команды `debug ip rip` включите трассировку на маршрутизаторе Router1. Посмотрите, как маршрутизаторы обмениваются маршрутной информацией, поясните выводимую информацию. Отключите трассировку с помощью команды `no debug ip rip`. Командой `no router rip` отключите RIP на всех маршрутизаторах, сохраните конфигурацию.

Настройка EIGRP на маршрутизаторах

Включите EIGRP на всех маршрутизаторах, образуя автономную систему с номером 100:

```
Router1#configure terminal
Router1(config)#router eigrp 100
Router1(config-router)#network 172.16.10.0
Router1(config-router)#network 10.1.1.0
Router1(config-router)#exit
```

```
Router1(config)#exit
Router1#write memory

Router2#configure terminal
Router2(config)#router eigrp 100
Router2(config-router)#network 10.1.1.0
Router2(config-router)#exit
Router2(config)#exit
Router2#write memory

Router3#configure terminal
Router3(config)#router eigrp 100
Router3(config-router)#network 172.16.10.0
Router3(config-router)#exit
Router3(config)#exit
Router3#write memory
```

Посмотрите изменение в конфигурации каждого маршрутизатора. Сеть 10.1.1.0/24 должна быть воспринята как сеть 10.0.0.0/8, а сеть 172.16.10.0/24 — как сеть 172.16.0.0/16. Это связано с классами IP адресов.

С помощью команды `show ip route` определите параметры, по которым работает протокол EIGRP. С помощью команды `show ip route` посмотрите таблицы маршрутов. С помощью команд `ping` и `traceroute` проверьте доступность других маршрутизаторов.

С помощью команды `debug eigrp packets` посмотрите, какими пакетами обмениваются маршрутизаторы. Остановите вывод этой информации командой `no debug eigrp packets`.

Командой `no router eigrp 100` отключите EIGRP на всех маршрутизаторах, сохраните конфигурацию.

Настройка OSPF на маршрутизаторах

Включите OSPF на всех маршрутизаторах, присвоив процессу номер 100 с областью действия номер 0:

```
Router1#configure terminal
Router1(config)#router ospf 100
Router1(config-router)#network 172.16.10.0 0.0.0.255 area 0
Router1(config-router)#network 10.1.1.0 0.0.0.255 area 0
Router1(config-router)#exit
Router1(config)#exit
Router1#show running-config
Router1#write memory
```

```
Router2#configure terminal
Router2(config)#router ospf 100
Router2(config-router)#network 10.1.1.0 0.0.0.255 area 0
Router2(config-router)#exit
Router2(config)#exit
Router2#show running-config
Router2#write memory

Router3#configure terminal
Router3(config)#router ospf 100
Router3(config-router)#network 172.16.10.0 0.0.0.255 area 0
Router3(config-router)#exit
Router3(config)#exit
Router3#show running-config
Router3#write memory
```

Посмотрите изменение в конфигурации каждого маршрутизатора.

С помощью команды `show ip route` определите параметры, по которым работает протокол OSPF. С помощью команды `show ip route` посмотрите таблицы маршрутов. С помощью команд `ping` и `traceroute` проверьте доступность других маршрутизаторов.

С помощью команд `ip ospf interface`, `show ip ospf database` и `debug ip ospf events` посмотрите все параметры протокола OSPF.

Сохраните конфигурацию.

9.4.6. Задание для самостоятельной работы

1. Загрузите проект из самостоятельной работы раздела 9.4.4 и пересохраните его (например, `lab04-04.pkt`).
2. Отключите на всех маршрутизаторах статическую маршрутизацию. Проверьте с помощью команды просмотра конфигурации маршрутизатора.
3. Настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу RIP. На каждом маршрутизаторе посмотрите таблицу маршрутизации. На каждом компьютере выполните команды трассировки `tracert` других компьютеров.
4. Отключите на маршрутизаторе Router1 последовательный интерфейс Serial 0. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотрите таблицу маршрутизации. Определите, через какую сеть будут маршрутизироваться пакеты? На каждом компьютере выполните команды трассировки `tracert` других компьютеров. Сохраните файлы конфигурации маршрутизаторов.
5. Отключите RIP и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу IGRP. На каждом маршрутиза-

торе посмотрите таблицу маршрутизации. На каждом компьютере выполните команды трассировки `tracert` других компьютеров.

6. Отключите на маршрутизаторе Router1 последовательный интерфейс Serial 0. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотрите таблицу маршрутизации. Определите, через какую сеть будут маршрутизироваться пакеты? На каждом компьютере выполните команды трассировки `tracert` других компьютеров. Сохраните файлы конфигурации маршрутизаторов.
7. Отключите IGRP и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу OSPF. На каждом маршрутизаторе посмотрите таблицу маршрутизации. На каждом компьютере выполните команды трассировки `tracert` других компьютеров.
8. Отключите на маршрутизаторе Router1 последовательный интерфейс Serial 0. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотрите таблицу маршрутизации. Определите, через какую сеть будут маршрутизироваться пакеты? На каждом компьютере выполните команды трассировки `tracert` других компьютеров. Сохраните файлы конфигурации маршрутизаторов.

Словарь терминов

Asynchronous Transfer Mode (ATM)
Мультисервисная, высокоскоростная технология асинхронной передачи данных (ячеек небольшого размера фиксированной длины — по 53 байта) со встроенной поддержкой обеспечения гарантированного качества обслуживания (QoS); может применяться при построении магистральных сетей, например, поверх SONET/SDH 19, 40, 116, 118–120, 216, 293, 317, 318, 320

Bluetooth
Технология, обеспечивающая беспроводную передачу данных на небольшие расстояниях между различными устройствами (например, мобильными персональными компьютерами, мобильными телефонами и другими устройствами) в режиме реального времени 76, 139, 141–145, 147

Click Modular Router
Специализированное программное обеспечение для создания высокопроизводительных программных маршрутизаторов 190, 191

Edge LSR (E-LSR)
Маршрутизатор на границе сети MPLS, осуществляющий классификацию поступающих в MPLS-сеть пакетов, их фильтрацию, управление трафиком и т.п. 211

ENUM
Сетевой протокол, определяющий выбор маршрутов для связи с различными устройствами, принадлежащими одному абоненту (пользователю телефонного номера в международном формате E.164), и устанавливающий соответствие между номером в формате E.164 (международный формат телефонных номеров, определяемый в Рекомендации E.164 ITU) и доменным именем (Domain Name System, DNS) 318

FDDI
Сеть и технология в виде волоконно-оптического маркерного кольца со скоростью передачи данных 100 Мбит/с 73, 76, 98, 99, 101, 102, 104

Frame Relay (FR)
Технология доставки сообщений в сетях передачи данных с коммутацией пакетов; может использоваться для управления пульсирующим трафиком между локальными сетями и территориальной сетью, а

(ТфОП); имеет свой стек протоколов с одноимённым названием 19, 34, 108, 109, 111

Автономная система
Совокупность сетей, находящихся под единым административным управлением 193, 199–201, 203–207, 210

Адрес 6to4
Используется узлами, работающими как с протоколом IPv4, так и с протоколом IPv6; формируется путём объединения префикса 2002::/16 с 32-битным адресом IPv4, в результате чего получается 48-битный префикс (например, для IPv4-адреса 131.107.0.1 префиксом адреса 6to4 является 2002:836B:1::/48) 173

Адрес группы интерфейсов (Multicast) IPv6
Идентифицирует множество интерфейсов IPv6 171, 174

Адрес запроса узлов (Solicited-node Multicast Address) IPv6
Совмещает в себе функции адреса одиночного интерфейса и адреса Anycast; состоит из префикса FF02::1:FF00:0/104 и последних 24 бит Unicast или Anycast-адреса IPv6 175

Адрес локальной подсети (Site-Local-Use Address) IPv6
Определяется префиксом формата 11111110 11, эквивалентен частному пространству адресов IPv4 (10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16) 171, 172

Адрес локальной связи (Link-Local-Use Address) IPv6
Определяется префиксом формата 11111110 10, используется при обмене данными между соседними узлами сети с единой связью; эквивалентны адресам APIPA (Automatic Private IP Addressing) в IPv4 (использующим префикс 169.254.0.0/16); необходим для процессов изучения окружения и всегда настраивается автоматически 171

Адрес любого интерфейса группы интерфейсов (Anycast)
Идентифицирует некоторую группу интерфейсов IPv6; используется для обмена данными по схеме «один–один из многих» с доставкой на один интерфейс 171, 175, 176

Адрес одиночного интерфейса (Unicast) IPv6
Идентифицирует только один интерфейс IPv6 171, 175–177

Алгоритм вектора расстояния (Distance-Vector)
Тип алгоритма маршрутизации, в котором маршрутизатор через заранее определённые промежутки времени передаёт соседним маршрутизаторам содержимое своей таблицы маршрутизации 193, 195

Алгоритм состояния канала (Link-State)
Тип алгоритма маршрутизации, в котором маршрутизатор передаёт информацию только об изменениях состояния системы 193, 199

Архитектура сети
Набор уровней и протоколов 10, 28, 37, 59, 75, 211, 213, 290, 299,

довательно «соединённых» каналов на время передачи информации (до разъединения соединения) 15, 19, 294, 301, 310, 317, 330, 331

Коммутация пакетов
Режим передачи сообщений, при котором сообщения разбиваются на пакеты ограниченного размера, причём канал передачи занят только во время передачи пакета и освобождается после её завершения 15, 19, 40, 108, 109, 111, 210, 211, 216, 284, 293, 294, 313, 317, 330

Коммутация сообщений
Режим передачи, включающий приём, хранение, выбор исходящего направления и дальнейшую передачу сообщений без нарушения их целостности 15

Коммутация ячеек
Режим передачи пакетов фиксированного размера 15, 118

Концентратор (Hub)
Многопортовый повторитель с автосегментацией, в котором сигнал, полученный от одной из подключённых к нему станций, транслируется на все его активные порты 58, 87, 95, 100, 104–106, 108

Концепция бесклассовой маршрутизации (CIDR)
Подход, при котором для определения границ между номером сети и номером узла используется расширенный сетевой префикс, что позволяет организовывать сети произвольного размера 161, 163, 169, 198, 207

Линия связи
Совокупность физической среды распространения сигналов и оборудования, формирующих специализированные каналы, имеющие определённые стандартные показатели: полосу частот, скорость передачи и т.п. 7

Локальная сеть (Local Area Network, LAN)
Сеть здания или организации 19, 38, 47, 58, 59, 66, 72–74, 108, 111, 115, 118, 120, 122, 127, 131, 136, 137, 145, 188, 314

Маршрутизатор (Router)
Сетевое устройство, осуществляющее связь разных типов сетей и обеспечивает доступ к глобальной сети, управляет трафиком на основе протокола сетевого уровня 58, 59, 105, 163, 192, 193

Маска подсети (Subnet Mask) IPv4
32-разрядное двоичное число, в разрядах расширенного префикса содержащая единицу, а в остальных разрядах — ноль 161–163, 182, 192, 198, 202

Метод Demand Priority (приоритетный доступ по требованию)
Детерминированный метод разделения общей среды в технологии 100VG-AnyLAN, использующий два уровня приоритетов: низкий — для обычных приложений и высокий — для мультимедийных приложений, чувствительных к задержкам 76, 106

Метрика (Metric)
Обобщённая характеристика качеств маршрута 193–195, 197, 198,

доступ к сетевым ресурсам 33, 42, 43

Пропускная способность канала
Максимально возможная при определённых условиях скорость, при которой информация может передаваться по конкретному тракту связи или каналу 40, 46, 54, 58, 59, 90, 92, 116, 119, 121, 131, 134, 136, 138, 284, 299

Протокол (сетевой протокол)
Набор правил, позволяющий осуществлять соединение и обмен данными между двумя включёнными в сеть компьютерами; набор правил, описывающих формат и назначение пакетов, которыми обмениваются одноранговые сущности 9, 31, 34–37, 39–41, 69–71, 76, 78, 80, 82, 85, 86, 96, 99, 103, 104, 109, 115, 118, 136, 143–145, 152, 155, 156, 164, 168, 177, 178, 185–187, 189, 190, 193, 195–203, 205–207, 209, 211, 215, 220–224, 228, 230–232, 234–236, 239, 293, 295–298, 310–314, 317–320, 322–324, 329, 332, 400, 403, 409

Расширенный сетевой префикс
Совокупность номера сети и номера подсети 156, 161–163, 169, 171, 176, 178

Региональная сеть (Metropolitan Area Network, MAN)
Сеть уровня города или региона 19, 74, 115

Сеансовый уровень (Session Layer) ISO/OSI
Реализует службу имён (отображение логических имён в сетевые адреса), устанавливает сеансы между службами и создаёт точки для контрольной синхронизации в случае потери связи 32, 35, 37, 39

Сервис или услуга (Service)
Набор примитивов, которые предоставляются вышележащему уровню нижележащим; описывает интерфейс между двумя уровнями, в котором нижележащий уровень является поставщиком услуги, а вышележащий — её потребителем 9, 10, 26, 115, 145, 179, 188, 206, 215, 216, 223, 292, 332, 334

Сетевая доступность
Диапазон времени сетевой достижимости между входной и выходной точкой сети 208

Сетевой коммутатор (Switch)
Сетевое устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного сегмента и хранящее в памяти таблицу MAC-адресов, в которой указывается соответствие MAC-адреса узла порту коммутатора 58, 59, 72, 88, 89, 100, 101, 105, 112, 120, 121

Сетевой уровень (Network Layer) ISO/OSI
Предоставляет средства установления, поддержания и разрыва сетевого соединения, а также функциональные и процедурные средства для обмена по сетевому соединению сетевыми сервисными блоками данных между транспортными логическими объектами 29, 35, 42,

операций, обеспечивающих надёжную и экономически эффективную передачу данных 31, 35, 37, 39, 79, 206, 207, 220, 222, 236, 240

Уплотнение с временным разделением (TDM)
Распределение каналов идёт по времени, т.е. каждый передатчик транслирует сигнал на одной и той же частоте, но в различные промежутки времени (как правило, циклически повторяющиеся) при строгих требованиях к синхронизации процесса передачи 123

Уплотнение с кодовым разделением (CDM)
Передача сигналов ведётся на одной и той же частоте, но каждый передатчик заменяет каждый бит исходного потока данных на CDM-символ — уникальную кодовую последовательность длиной в 11, 16, 32, 64 и т.п. бит 124

Уплотнение с пространственным разделением
Основано на разделении сигналов в пространстве, когда передатчик посылает сигнал, используя код c, время t и частоту f области si, т.е. есть передача данных ведётся только в границах определённой территории, на которой любому другому устройству запрещено передавать свои сообщения 123

Уплотнение с частотным разделением (FDM)
Каждое устройство работает на определённой частоте, благодаря чему несколько устройств могут вести передачу данных на одной территории 123

Уровень ошибок
Частота появления ошибок (ошибкой считается приём 1 при переданном 0 и наоборот) 46, 107

Уровень представления (Presentational Layer) ISO/OSI
Устанавливает способы представления информации, которой обмениваются прикладные логические объекты или на которую они ссылаются в процессе этого обмена 32, 37

Уровень управления логическим каналом (LLC)
Отвечает за передачу кадров между узлами с различной степенью надёжности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем 35, 38, 40, 73, 78–80

Физический уровень (Physical Layer) ISO/OSI
Обеспечивает передачу битовых потоков без каких-либо изменений между логическими объектами уровня звена данных по физическим соединениям 28, 29, 37, 42, 72, 74–76, 108

Цифровая сеть с интеграцией служб (ISDN)
Сеть с коммутацией каналов (телефонная сеть), обеспечивающая полностью цифровые соединения между оконечными устройствами для поддержания широкого спектра информационных услуг 40, 43, 47, 76, 112, 293, 295, 310

Ширина полосы
Ширина полосы передаваемого сигнала, ограничиваемая передат-

Список иллюстраций

Список таблиц

Используемая литература

1. *Кулябов Д. С.*, *Королькова А. В.* Архитектура и принципы построения современных сетей и систем телекоммуникаций. — М. : РУДН, 2008.
2. Современные концепции управления инфокоммуникациями / К. Е. Самуйлов, Д. С. Кулябов, А. В. Королькова, Ю. В. Гайдамака, И. А. Гудкова, П. О. Абаев. — М. : РУДН, 2013.
3. *Васин Н. Н.* Построение сетей на базе коммутаторов и маршрутизаторов. — Интуит, 2011.
4. *Таненбаум Э.* Компьютерные сети. — Пятое. — Питер : Питер, 2013.
5. *Семёнов А. Ю.* Протоколы Интернет. Энциклопедия 2-е изд. — 2005.
6. *Семёнов А. Ю.* Алгоритмы телекоммуникационных сетей. — Изд-во Интернет-университет информационных технологий, Бином, 2007.
7. *Олифер В. Г.*, *Олифер Н. А.* Основы сетей передачи данных. Курс лекций. — Издательство: Интернет-университет информационных технологий, Бином, 2005.
8. *Олифер В. Г.*, *Олифер Н. А.* Компьютерные сети. Принципы, технологии, протоколы. Учебник для ВУЗов 3-е издание. — Питер : Питер, 2007.
9. *Самуйлов К. Е.* Методы анализа и расчёта сетей ОКС 7: Монография. — М. : Изд-во РУДН, 2002.
10. *Иванцов И.* Стеки протоколов // Журнал сетевых решений LAN. — 2007. — № 3.
11. *Самуйлов К. Е.*, *Галентовская М.* Введение в систему сигнализации No 7 // Сети. — 1999. — 8–9. — URL: `http://www.osp.ru/nets/1999/08-09/144246/_p1.html`.
12. *Никольский Н. Н.* Передача ОКС7 через IP // Сети и системы связи. — 2005. — № 7. — URL: `http://www.ccc.ru/magazine/depot/05_07/0301.htm`.
13. *Королькова А. В.*, *Кулябов Д. С.* Сетевые технологии. Лабораторные работы: учебное пособие. — Москва : РУДН, 2014. — ISBN 978-5-209-05606-5.
14. Радиопередающие устройства / В. В. Шахгильдян, В. Б. Козырев, А. А. Ляховкин, В. П. Нуянзин, В. М. Розов, М. С. Шумилин ; под ред. В. В. Шахгильдяна. — 3-е изд. — Радио и связь, 2003. — ISBN 5-256-01237-1.

15. *Шахгильдян В. В.*, *Шумилин М. С.*, *Козырев В. Б.* Проектирование радиопередатчиков / под ред. В. В. Шахгильдяна. — 4-е изд. — Радио и связь, 2000. — С. 656. — ISBN 5-256-01378-5.

16. *Лакнер Х.* Мобильность и полоса пропускания. Обзор актуальных стандартов IEEE 802 за последний год // LAN. — 2007. — № 6. — URL: http://www.osp.ru/lan/2007/06/4238886/.

17. Сети Fast Ethernet // Оптилинк. — 2001. — URL: http://www.optilink.ru/Techdoc/Ethernet/fast_ethernet.htm.

18. *Смелянский Р. Л.* Системы передачи данных и сети ЭВМ. — URL: http://www.kgtu.runnet.ru/WD/TUTOR/cn/index.html.

19. Основы локальных сетей. — URL: http://www.intuit.ru/department/network/baslocnet/.

20. *Кунегин С. В.* Общее описание метода информационного обмена в сетях передачи данных Frame Relay. — URL: http://kunegin.narod.ru/ref/fro/index.htm.

21. *Филимонов А.* Сети Frame Relay // Сети ЭВМ и телекоммуникации (курс лекций). — URL: http://lectures.net.ru/lectures/.

22. *Мельников Д.* Frame Relay для профессионалов и не только // Сети. — 1997. — № 10. — URL: http://www.osp.ru/nets/1997/10/142934/.

23. *Андронов С.* О структуре и свойствах современных пакетных сетей // JetInfo. — 1999. — 6(73). — URL: http://www.jetinfo.ru/1999/6/1/article1.6.1999.html.

24. *Афонцев Э.* Metro Ethernet. Архитектура и технологии // NAG.ru. — 2005. — URL: http://www.nag.ru/2005/0227/0227.shtml.

25. *Афонцев Э.* Назад в будущее. Metro Ethernet // NAG.ru. — 2005. — URL: http://www.nag.ru/2005/0212/0212.shtml.

26. Беспроводные сети Wi-Fi / А. В. Пролетарский, И. В. Баскаков, Р. А. Федотов, А. В. Бобков, Д. Н. Чирков, В. А. Платонов. — Изд-во «Интернет-университет информационных технологий — ИНТУИТ.ру», БИНОМ, 2007. — URL: http://www.intuit.ru/department/network/wifi/.

27. *Гулевич Д. С.* Сети связи следующего поколения. — Открытые системы, ИНТУИТ, 2007. — URL: http://www.intuit.ru/department/network/ndnets/.

28. *Duffy J.* Verizon Wireless leads group offering IMS extensions // NetworkWorld.com. — 2006. — URL: http://www.networkworld.com/news/2006/072706-verizon-wireless-ims.html.

29. *Duffy J.* Verizon, others offer IMS extensions: Now comes the hard part for A-IMS // NetworkWorld.com. — 2006. — URL: http://www.networkworld.com/news/2006/073106-a-ims.html.

30. *Хисматуллин И.* IMS предлагается дополнить // Сети. — 2006. — № 14. — URL: http://www.osp.ru/nets/2006/14/3199456/.

31. *A J.* Packet Tracer Network Simulator. — Packt Publishing, 2014. — ISBN 9781782170433.

Рекомендации для чтения

Многие темы не были рассмотрены во всей полноте и со всеми подробностями, а многие вопросы были опущены из-за недостатка места. В этой главе предлагается список дополнительной литературы для читателей.

1. *Таненбаум Э., Уэзеролл Д.* Компьютерные сети. 5-е изд. — СПб.: Питер, 2012. — 960 с.: ил. — ISBN 978-5-459-00342-0.
   В книге последовательно изложены основные концепции компьютерных сетей. Авторы подробнейшим образом объясняют устройство и принципы работы аппаратного и программного обеспечения. Изложены как теоретические принципы, так и примеры функционирования Интернета и компьютерных сетей различного типа. Освещаются такие современные темы как беспроводные сети, сети 3G, пиринговые сети, потоковое вещание, интернет-телефония. Рекомендуется для вдумчивого теоретического изучения предмета.
2. *Олифер В. Г., Олифер Н. А.* Компьютерные сети. Принципы, технологии, протоколы. 4 изд. — СПб.: Питер, 2010. — 916 с.
   Книга наполнена теоретическим материалом, однако содержит и элементы практических знаний. Наряду с книгой Таненбаума эту книгу можно отнести к энциклопедии сетевых технологий.
3. *Столлингс В.* Современные компьютерные сети. 2-е изд. — СПб.: Питер, 2003. — 783 с.: ил.
   Книга посвящена современным аспектам развития высокоскоростных объединённых TCP/IP и ATM сетей. В ней рассматривается широкий круг вопросов: от обработки одиночного пакета или ячейки в очереди на маршрутизаторе или коммутаторе до универсальных методов резервирования сетевых ресурсов для определённого типа трафика; от определения характеристик потока данных до способов их сжатия, позволяющих снизить нагрузку на сеть.
4. *Вишневский В. М., Портной С. Л., Шахнович И. В.* Энциклопедия WiMAX. Путь к 4G. — М.: Техносфера, 2009. — 472 с.
   Хотя содержание книги очевидно из заглавия, не стоит думать, что она только про технологию WiMAX. Её стоит использовать как фундаментальное руководство по беспроводным сетям вообще.
5. *Гольдштейн Б. С., Соколов Н. А., Яновский Г. Г.* Сети связи. — СПб.: БХВ Санкт-Петербург, 2010. — 400 с.
   Книга является учебником по современным сетям связи. В учебнике рассматриваются три сети, создававшиеся для поддержки следующих видов обслуживания: фиксированная телефонная связь, мобильные коммуникации и документальная электросвязь. Для каждой из трех сетей изложены идентичные по своему характеру базовые принципы построения и функционирования. Сформулирована основная цель дальнейшего развития трех рассматриваемых

сетей — переход к сети связи следующего поколения, известной по аббревиатуре NGN (Next Generation Network).

6. *Семёнов А. Б.* Администрирование структурированных кабельных систем. — НОУДПО «Институт АйТи». — М.: ДМК Пресс; М.: Компания АйТи, 2008. — 192 с.: ил.
   Рассмотрено состояние стандартизации в области администрирования информационных структурированных кабельных систем. Описана структура БД, используемой для описания текущей конфигурации проводки и планирования работ по ее изменению. Представлены стандартизованные схемы и правила формирования маркирующих индексов, а также составления записей для различных компонентов. Выполнен обзор программных и аппаратных средств, а также проектных приёмов, применение которых увеличивает эффективность текущей эксплуатации СКС. Затронуты вопросы эксплуатационного обслуживания СКС.
7. *Деарт В. Ю.* Мультисервисные сети связи. Ч. 1: Транспортные сети и сети доступа. — М.: Инсвязьиздат, 2007. — 166 с.; *Деарт В. Ю.* Мультисервисные сети связи. Ч. 2: Протоколы и системы управления сеансами (Softswitch/IMS). — М.: Брис-М, 2011. — 198 с.
   Книга написана по материалам курса лекций «Мультисервисные сети связи», прочитанного автором студентам, магистрам и аспирантам МТУСИ. В первую часть пособия вошли лекции по транспортным сетям и сетям доступа. Во второй книге рассматривается уровень управления сеансами сетей NGN. Уделяется большое внимание протоколам управления и сигнализации, которые реализуются в программных коммутаторах (Softswitch) и подсистемах мультимедийной связи (IMS).

Содержание

Учебное издание

**К. Е. Самуйлов, И. А. Шалимов, Н. Н. Васин,
В. В. Василевский, Д. С. Кулябов,
А. В. Королькова**

Сети и системы передачи информации: телекоммуникационные сети

Учебник

Технический редактор
Дизайн обложки

Издание подготовлено в авторской редакции
Компьютерная вёрстка *А. В. Королькова*, *Д. С. Кулябов*

Подписано в печать ___.___.2015 г. Формат 60×84/16. Печать офсетная.
Усл. печ. л. ___. Тираж экз. Заказ № ___.