Н. В. Будылдина, В. П. Шувалов

Сетевые технологии высокоскоростной передачи данных

Под ред. профессора В. П. Шувалова

Рекомендовано УМО по образованию в области Инфокоммуникационных технологий и систем связи в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению 11.03.02 – «Инфокоммуникационные технологии и системы связи» квалификации (степени) «бакалавр» и «магистр»

Москва
Горячая линия – Телеком
2018

УДК 621.396.2
ББК 32.884
Б90

Р е ц е н з е н т ы : доктор техн. наук, профессор *Л. Г. Доросинский*; доктор техн. наук, профессор *Д. Г. Неволин*

Будылдина Н. В., Шувалов В. П.

Б90 Сетевые технологии высокоскоростной передачи данных. Учебное пособие для вузов / Под ред. профессора В. П. Шувалова. – М.: Горячая линия – Телеком, 2018. – 342 с.: ил.

ISBN 978-5-9912-0536-8.

В компактной форме изложены вопросы построения инфокоммуникационных сетей, обеспечивающих высокоскоростную передачу данных. Представлены разделы, которые необходимы для понимания того как можно обеспечить передачу не только с высокой скоростью, но и с другими показателями, характеризующими качество предоставляемой услуги. Приведено описание протоколов различных уровней эталонной модели взаимодействия открытых систем, технологий транспортных сетей. Рассмотрены вопросы передачи данных в беспроводных сетях связи и современные подходы, обеспечивающие передачу больших массивов информации за приемлемые отрезки времени. Уделено внимание набирающей все большую популярность технологии программно-конфигурируемых сетей.

Для студентов, обучающихся по направлению подготовки «Инфокоммуникационные технологии и системы связи» квалификации (степени) «бакалавр» и «магистр». Книга может быть использована для повышения квалификации работниками электросвязи.

ББК 32.884

Адрес издательства в Интернет www.techbook.ru

Будылдина Надежда Вениаминовна, **Шувалов** Вячеслав Петрович
Сетевые технологии высокоскоростной передачи данных
Учебное пособие для вузов

Введение

Выступая в 2010 г. на конференции Technomy, председатель совета директоров компании Google Эрик Шмидт сказал: «Пять экзабайт информации создано человечеством с момента зарождения цивилизации до 2010 года, столько же сейчас создается каждые два дня...». Напомним, что 1 экзабайт = 1024 петабайта, 1 петабайт = 1024 терабайта, а 1 терабайт = 1024 гигабайта. Если использовать среднестатистическое Интернет-соединение для передачи всего 10 Тбайт данных из Беркли (США, Калифорния) в Сиэтл (США, Вашингтон), то потребуется 45 дней, в то время как пересылка винчестера с помощью курьерской службы занимает менее одного дня [1].

Понятно, что столь плачевные результаты обусловлены большим объемом передаваемой информации и относительно низкой скоростью передачи данных.

Что такое низкая, средняя, высокая скорость? Если открыть учебник [2], изданный в 1990 г., то там можно прочитать, что низкая скорость — это 300 бит/с, средняя — 600...1200 бит/с, а высокая — 9600 бит/с и более. Сегодня скорость 300 бит/с и даже более высокие, такие как 9600 бит/с, используются чрезвычайно редко. В локальных сетях скорости достигают нескольких Гбит/с, в городских они несколько меньше, и еще меньше скорости, которые достигаются в глобальных сетях. Таким образом, чем меньше расстояния, на которые передается информация, тем выше скорость. Итак, подчеркнем еще раз: высокая скорость есть понятие относительное, зависящее от того, в какой период времени мы рассматриваем эту характеристику сети передачи данных.

Далее поясним, пользуясь соответствующими первоисточниками, термины, входящие в состав названия учебного пособия «Сетевые технологии высокоскоростной передачи данных». Итак, «данные: информация, представленная в виде, подходящем для автоматической обработки ее автоматическими средствами при возможном участии человека» [3]. В современном толковом словаре изд. «Большая Советская Энциклопедия» сказано: «данные» в информатике — это информация, представленная в формализованном виде, что обеспечивает возможность ее хранения, обработки и передачи. Термин «данные» происходит от слова «data» (факт), а «информация» («information») означает разъяснение, изложение, т.е. сведения.

Слово «технология» происходит от греческого *τεχνη*, что означает «искусство», и *λογος* — «наука, учение». В толковом словаре С.И. Ожегова и Н.Ю. Швецовой «технология — это совокупность производственных методов и процессов в определенной отрасли про-

изводства, а также научное описание способов производства». Для информационных технологий первоначальным «сырьем» и конечной «продукцией» является информация, поэтому процессы преобразования информации можно назвать технологией.

Подводя итоги сказанного, отметим, что перед нами стоит задача рассмотреть различные методы преобразования информации, обеспечивающие передачу данных с высокой скоростью или передачу больших объемов информации (Big Data) за приемлемое время. Разумеется, при этом необходимо обеспечить требования к другим показателям, определяющим понятие QoS (качество услуг). Подчеркнем еще раз: высокая скорость понятие относительное, подверженное время от времени пересмотру.

В учебном пособии рассмотрен комплекс вопросов, относящихся к проблематике высокоскоростной передачи данных.

В главе 1 представлены основные понятия и определения, значение которых необходимо и достаточно для чтения и понимания последующего материала. Здесь затронуты вопросы обеспечения высокой скорости за счет использования соответствующей физической среды, рассмотрены ограничения, вызванные наличием помех в канале, показана возможность повышения скорости передачи информации за счет использования многопозиционных сигналов.

Сама по себе высокая скорость передачи информации еще недостаточна для того, чтобы удовлетворить потребности клиента, необходимо также обеспечить такие качественные показатели, как достоверность, надежность, минимальное время задержки и др. Эти вопросы рассмотрены в главе 2.

В главе 3 представлены технологии, обеспечивающие сегодня требования к высокоскоростной передаче данных в локальных сетях.

4-я глава посвящена описанию протоколов канального уровня, а 5-я — описанию сетевого и транспортного уровня модели взаимодействия открытых систем.

В главе 6 рассматривается четырехуровневая наложенная транспортная сеть и пути сокращения числа уровней с переходом на оптические транспортные сети. Здесь представлено описание таких технологий как ATM, SDH, MPLS-TP, PBB-TE.

В главе 7 рассмотрены беспроводные технологии высокоскоростной передачи данных (Wi-Fi, WiMAX, LTE).

Материал 8-й главы содержит результаты исследований, выполненных в Германии в лаборатории FILA (Future Internet Lab. Anchalt) под руководством проф. Э. Сименса.

Здесь рассмотрены основные препятствия, ограничивающие пропускную способность каналов, и некоторые соображения относитель-

но того, как решать проблему передачи больших данных (Big Data) в современных IP-сетях, приведены результаты экспериментальных исследований скорости передачи данных от величины задержки и процента потери пакетов.

В приложении к пособию дано краткое изложение принципов построения программно-конфигурируемых сетей, которые завоевывают в последнее время все большую и большую популярность. Дано описание технологии виртуализации сетевых функций NFV (Network Function Virtualization), приведено сравнение SDN и NFV.

Помимо авторов, указанных на титульном листе, в подготовке материалов для данного пособия приняли участие доцент СибГУТИ С.В. Тимченко (глава 7) и аспиранты СибГУТИ А.Ю. Бахарев и Д.С. Качан (глава 8).

Авторы считают своим приятным долгом выразить благодарность студенткам СибГУТИ А. Волковой и Е. Курносовой за помощь в подготовке разделов рукописи к печати.

Список литературы к введению

1. *Armbrust M., et. al.* Above The Clouds: A Berkeley View of Cloud Computing. 2009. Tec. Rep. No UUCB/EECS-2009-28.
2. *Шувалов В.П., Захарченко Н.В., Шварцман В.О. и др.* Передача дискретных сообщений: Учебник для вузов / Под ред. В.П. Шувалова. — М.: Радио и связь, 1990.
3. ГОСТ Р 53728-2009. Качество услуги «Передача данных». Дата введения — 2011–01–01. — М., 2011. — 6 с.

Глава 1

Основные понятия и определения

1.1. Информация, сообщение, сигнал

Рассмотрим цепочку взаимосвязанных понятий, которая представлена на рис. 1.1

Рис. 1.1. Цепочка понятий

Существует несколько определений понятия информации. Наиболее краткое, принадлежащее Н.Винеру, звучит так: «Информация есть информация, а не материя и не энергия». Это определение близко по смыслу к другому, более полному определению: «Информация — это третья составляющая трех основ мироздания (материя, энергия и информация), т.е. чтобы сделать автомобиль и т.п. надо иметь знания — это и есть информационная составляющая».

Наиболее часто встречающееся в учебниках определение звучит так: «Информация — это сведения о каких-либо процессах, событиях, фактах или предметах».

В математической теории (в теории связи) понятие «информация» носит объективный характер и определяется только для случайных событий. *Информация — это то, что уменьшает неопределенность события*.

Формой представления информации является сообщение, служащее для передачи ее от источника к получателю. Примером сообщений являются текст телеграммы, речь, музыка, чертеж и др.

Материальным носителем или физическим процессом является сигнал, несущий (отражающий) передаваемое сообщение.

Итак, в сообщениях заложена информация. Поэтому естественно возникает задача оценить информативность этих сообщений, т.е. количество информации, заложенное в них.

Очевидно, что из трех определений информации для оценки количества информации, содержащегося в сообщении, можно использовать только четвертое. Информация — это то, что уменьшает неопределенность события, т.е. чем выше неопределенность события, тем большее количество информации несет принимаемое сообщение.

Рассмотрим работу реле. Пусть поступление тока в обмотку реле вызывает замыкание контактов. Итак, пусть контакты находятся в замкнутом состоянии с вероятностью p_1, в разомкнутом — с вероятностью $p_2 = 1 - p_1$. Варианты значений p_1 приведены в табл. 1.1.

Таблица 1.1. Вероятности событий и оценка количества информации

N	p_1	p_2	Количество информации
1	1	0	$J=0$
2	0,7	0,3	$J>0$
3	0,5	0,5	$\max J$
4	0,3	0,7	$J>0$
5	0	1	$J=0$

В вариантах 1 и 5 неопределенность отсутствует. Первый вариант соответствует случаю, когда контакты всегда замкнуты, и мы об этом знаем (нам известна вероятность этого события). Вариант 5 соответствует случаю, когда контакты всегда разомкнуты (вероятность этого события также равна 1, и мы об этом знаем). Очевидно, что количество информации J, которое несет сообщение как в варианте 1, так и в 5, равно нулю ($J=0$).

Максимальное количество информации, которое можно получить, соответствует максимальной неопределенности (вариант 3).

Итак, в основе определения количества информации, содержащегося в сообщении, лежит вероятностный подход, учитывающий степень неожиданности (новизны) сообщения. Что это значит? Предположим, произошло какое-то явление или событие. Чем оно более редкое, тем сообщение о нем для нас неожиданнее, информативнее и интереснее. Наоборот, те сообщения, которые для нас оказываются обыденными, привычными, интереса не представляют и оказываются малоинформативными. Например, сообщение о том, что зимой солнце восходит позже, чем летом, для нас никакой информации не несет. Естественно, что подобные сообщения нет смысла передавать. Но если сообщили, что студенческая стипендия будет равна зарплате президента России, то в силу большой неожиданности это сообщение будет содержать очень большое количество информации (и довольно приятной). Вероятность появления такого сообщения чрезвычайно мала (стремится к нулю).

Количество информации в отдельно взятом сообщении определяется величиной, обратной вероятности сообщения, вычисленной в логарифмических единицах:

$$J(a)=\log_k \frac{1}{p(a)} = -\log_k p(a),$$

где $p(a)$ — вероятность сообщения a; k — основание логарифма. При $p(a)=1$ количество информации равно нулю, т.е. сообщение об известном событии никакой информации не несет.

Количество информации, содержащееся в нескольких независимых сообщениях, равно сумме количества информации в каждом из них. Это соответствует и интуитивным представлениям об увеличе-

нии информации при получении дополнительных сообщений.

Как выбирается основание логарифма? Оно может быть любым. Но в теории и технике связи наиболее употребительным является основание 2, так как по каналам связи чаще всего передаются так называемые двоичные сообщения, у которых элементы могут иметь одно из двух значений: 0 или 1, а также 1 или -1. Кроме того, двоичная система счисления используется в электронно-вычислительных машинах.

Если принять основание логарифма равным двум, тогда количество информации, содержащейся в сообщении, выражается в двоичных единицах:

$$J(a) = -\log_2 p(a).$$

Двоичную единицу обычно называют битом — от «binary digit» (двоичная цифра).

Совокупность всех возможных сообщений и вероятностей их появления образует ансамбль сообщений. Если ансамбль состоит всего из двух сообщений a_1 и a_2 (например, вида «да» и «нет» или 0 и 1), которые являются независимыми и равновероятными, т.е. $p(a_1) = p(a_2) = \frac{1}{2}$, то каждое из сообщений несет одну двоичную единицу (один бит) информации:

$$J = -\log_2 p(a_1) = -\log_2 p(a_2) = -\log_2 \frac{1}{2} = 1 \text{ бит}.$$

Определим количество информации в слове из n букв, если алфавит состоит из m букв (32 в русском алфавите) и вероятности букв одинаковы, а сами буквы следуют независимо друг от друга. Количество информации при передаче одной буквы $J(a_i) = -\log_2 p(a_i)$. Так как мы приняли вероятности появления букв одинаковыми, то $p(a_i) = \frac{1}{m}$ и количество информации, содержащееся в любой букве,

$$J(a_i) = -\log_2 \frac{1}{m} = \log_2 m.$$

Если считать, что буквы следуют независимо, то количество информации в слове из n букв:

$$J_{\text{сл}} = \sum_{i=1}^{n} J(a_i) = n \log_2 m.$$

Например, для слова «студент» ($n = 7$, $m = 32$) количество информации составит 35 бит.

Часто возникает потребность оценивать информационные свойства источника сообщений в целом. Такой характеристикой является среднее количество информации, приходящееся на одно сообщение — энтропия источника сообщений.

Выше рассмотрен пример, когда появление разных букв было рав-

новероятным. Реальные сообщения имеют алфавиты с более сложной вероятностной структурой. Возьмем, например, алфавит русского языка. Оказывается, его буквы встречаются в тексте неодинаково часто — имеют различную вероятность появления. Например, буквы О и Е появляются в тексте гораздо чаще, чем буквы Ц и Щ.

Реальные сообщения имеют еще одну особенность. Их элементы не только разновероятны, но еще и связаны друг с другом определенными закономерностями. Например, в любом осмысленном тексте появление гласной буквы влечет за собой со значительной вероятностью появление согласной, т.е. менее вероятны двухбуквенные сочетания гласных или согласных. А их трехбуквенные сочетания встречаются еще реже. (Напомним, что сочетание «*ееe*» встречается только в одном слове русского языка. Каком?)

Указанные закономерности вызваны особенностями структуры языка, они существуют незримо, мы их просто в силу привычки не ощущаем. Но эти закономерности объективно существуют, и их оценивают статистическими (вероятностными) связями. В общем случае они довольно сложны, и мы не будем останавливаться на их количественной оценке.

Отметим только, что наличие статистических связей между элементами сообщений уменьшает их информативность, энтропию.

Наличие таких структурных свойств сообщений, как разновероятность элементов, их статистическая связь, приводит к избыточности реальных сообщений. Что это значит? То, что фактически в сообщениях «вмещается» меньше информации, чем могло бы быть, если бы элементы оказались равновероятными и независимыми. Их фактическая информативность оказывается меньше потенциально возможной.

За счет избыточности в реальных сообщениях много «воды». Это значит, что необходимые сведения, информацию можно было бы передать более короткими сообщениями, с меньшим количеством элементов. Следовательно, избыточность — это плохо? Не всегда. Преподаватель, выделяя важное место в лекции, иногда повторяет сказанное. Студент, конспектируя лекцию, прибегает к различным, понятным ему сокращениям слов и даже фраз. И потом без особого труда восстанавливает смысл записанного. Без избыточности он не смог бы это сделать: сокращенная часть информации оказалась бы потерянной.

Прочтите фразу: «Поздр--ляю с наст-п--щим Н-в-м го-ом». Фраза выглядит не очень привлекательно, но ее смысл понятен только благодаря избыточности русского языка.

С другой стороны, избыточность вредна там, где необходимо передать информацию с максимальной эффективностью. Например, пе-

редать по каналу связи информацию с наибольшей скоростью, эффективно используя его пропускную способность. Если не принять специальных мер по устранению избыточности, то передача информации займет слишком много времени из-за нерациональной насыщенности ею реальных сообщений.

Оказывается, что избыточность письменной речи в русском языке близка к 0,5. Это означает, что примерно 50% букв текста не несут никакой информации, являются «балластом». Еще большая избыточность наблюдается в английском языке. В то же время в иврите количество гласных сведено до минимума, что привело к уменьшению избыточности.

Есть ли способы борьбы с избыточностью там, где она нежелательна или даже вредна? Есть. Этим занимается специальное направление в теории информации и связи, использующее различные способы кодирования и преобразования сообщений. Примером может служить статистическое кодирование, при котором наиболее часто встречающиеся элементы сообщения кодируются (представляются) более коротким сигналами. Для передачи же более редких элементов используются длинные сигналы. Такой принцип реализован в коде Морзе. Самая частая буква в английском языке — буква E — передается самым коротким сигналом — точкой. Сравнительно редко появляющаяся буква Q кодируется длинным сочетанием точек и тире.

Суть такого кодирования заключается в следующем. Если по какой-то причине при передаче «потеряется» буква E (например, из-за воздействия помех), то она сравнительно легко интуитивно «восстановится»: вследствие высокой частоты ее появления к ней просто привыкают. Редкую же букву необходимо «защитить» более длинным сигналом, так как интуитивно ее «восстановить» значительно труднее.

1.2. Скорость передачи информации

Рассмотрим это понятие на примере передачи цифровых сигналов вида, представленного на рис. 1.2, где $U(t)$ принимает два значения — 0 и 1.

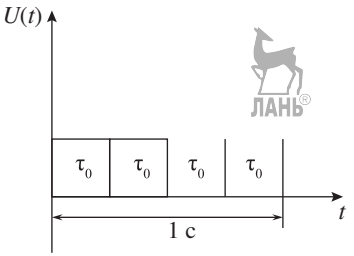


Рис. 1.2. Цифровой сигнал

Импульсы, изображенные на рис. 1.2, называют *единичными элементами*. Число единичных элементов, переданных в секунду, называют *скоростью телеграфирования* или *скоростью модуляции* B. Скорость телеграфирования измеряется в Бодах. Очевидно, что

$$B = \frac{1}{\tau_0}.$$

В соответствии с теоремой Найквиста,

$$B \leqslant 2\Delta F, \tag{1.1}$$

где ΔF — полоса частот линии, в которой осуществляется передача двоичных цифровых сигналов.

Таким образом, в стандартном телефонном канале скорость телеграфирования не может превышать 6200 Бод, так как $\Delta F = 3100$ Гц.

Для случая, представленного на рис. 1.2, $B = 4$ Бода. Помимо понятия «скорость телеграфирования» различают понятие «скорость передачи информации R». Скорость передачи информации измеряется, как известно, в битах в секунду. Если каждый единичный элемент «несет» на себе один бит, то $R = B$. Возможны случаи, когда $R < B$ и $R > B$. Начнем рассмотрение с варианта, когда $R < B$. Рассмотрим передачу некоторой последовательности 0 и 1 (0 — бестоковая посылка, 1 — токовая) (рис. 1.2). Пусть каждый единичный элемент повторяется два раза (см. рис. 1.3).

Так как за 2 с было теперь передано 8 единичных элементов, то по-прежнему $B = 4$ Бода. Повторение не увеличивает количество информации. По-прежнему мы имеем 4 бита, но на интервале 2 с. Следовательно, число бит/с будет только 2. Итак, в нашем случае $R = B/2$.

Может ли быть $R > B$? Рассмотрим случай, когда от источника сообщений поступает последовательность 1011001101. Пусть каждый из передаваемых элементов «несет» на себе 1 бит.

Объединим элементы последовательности попарно: 10 11 00 11 01. Вариантов пар, очевидно, будет четыре. А именно: 00, 01, 10, 11.

Для того, чтобы можно было различить на приеме сигналы, соответствующие этим парам, будем передавать эти пары при помощи импульсов длительности τ_0, но с различной амплитудой $00 \to 0$ В, $01 \to 1$ В, $10 \to 2$ В, $11 \to 3$ В (см. рис. 1.4).

В этом случае число элементов, переданных в секунду (рис. 1.4), будет равно 5, а скорость передачи информации $R = 10$ бит/с, так как каждый из единичных элементов (рис. 1.4) несет на себе 2 бита.

Соотношение $R = 2B$ реализуется в технологии HDSL (High Digital Subscriber Line). Здесь применяется преобразование сигналов, поступающих от источника с использованием кода 2B1Q (Two-Binary, One Quaternary). Сущность кодирования 2B1Q заключается в преоб-

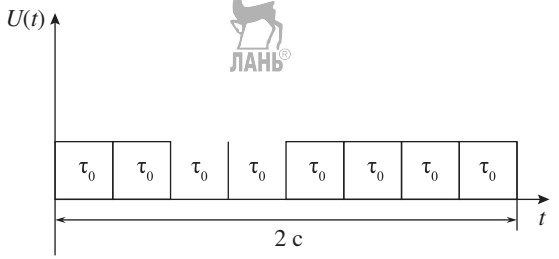


Рис. 1.3. Передача каждого единичного элемента два раза

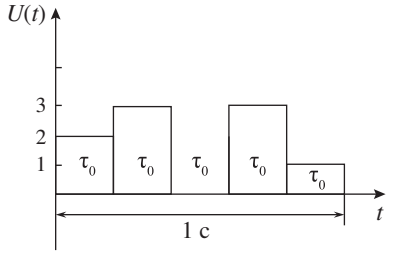


Рис. 1.4. Передача цифровых сигналов со скоростью $R = 2B$

разовании двух двоичных единичных элементов в один четверичный (табл. 1.2). Таким образом, код 2B1Q обеспечивает передачу на одном единичном интервале двух бит.

Таблица 1.2. Формирование кода 2B1Q

$B^{(2)}$	10	11	01	00
$Q^{(1)}$	+3	+1	−1	−3

Вид сигнала, соответствующий передаче последовательности 100001101, представлен на рис. 1.5, а спектр сигнала для скорости передачи информации 2320 Кбит/с — на рис. 1.6.

Из рис. 1.6 видно, что максимум энергетического спектра приходится на низкие частоты, в спектре содержится постоянная составляющая. Все это делает сигнал 2B1Q достаточно чувствительным к искажениям и помехам.

Желание обеспечить более высокую скорость передачи данных вызывают необходимость передачи на одном единичном интервале трех и более бит ($J \geqslant 3$). Количество уровней при этом для $J = 3$ будет равно 8, для $J = 4$ уже имеем 16 уровней. Обозначим число уровней буквой M, тогда

$$B \leqslant 2\Delta F \log_2 M. \qquad (1.2)$$

В общем виде M в формуле (1.2) — это количество различимых состояний информационного параметра сигнала. Так, если речь идет

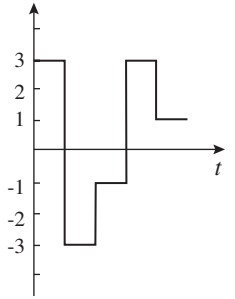


Рис. 1.5. Вид сигнала при кодировании кодом 2B1Q

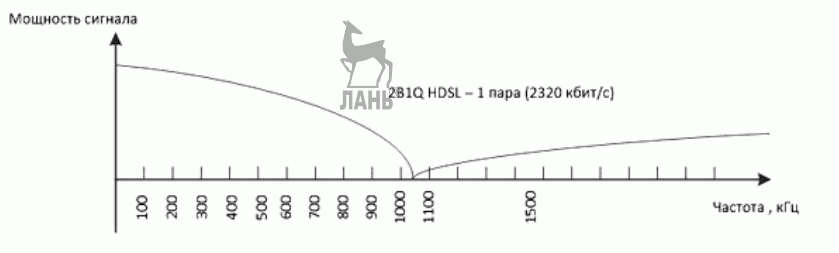


Рис. 1.6. Спектр сигнала для скорости передачи информации 2320 Кбит/с

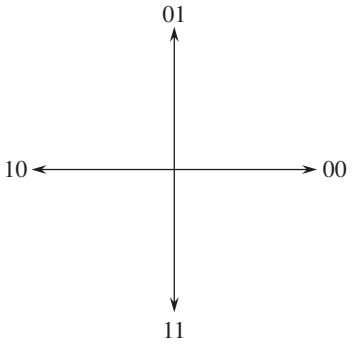


Рис. 1.7. Один единичный элемент «несет» на себе 2 бита. Число различимых фаз сигнала 4

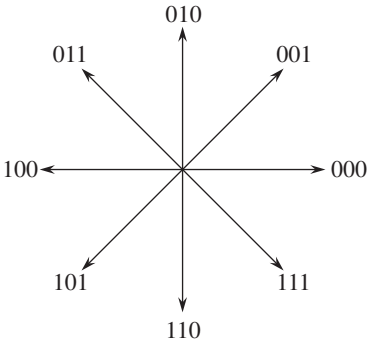


Рис. 1.8. Один единичный элемент «несет» на себе 3 бита. Число различимых фаз сигнала 8

о передаче с использованием фазовой модуляции, то это число различимых фаз сигнала.

Если максимальное значение уровня сигнала, передаваемого по линии, ограничено (а оно всегда ограничено), то для $U_{\max} = 1$ В при $J = 3$ имеем минимальное расстояние по уровню между цифровыми сигналами 0,125 В, для $J = 4$ — уже 0,0625 В. Следовательно, по мере роста скорости передачи информации помехоустойчивость, т.е.

способность противостоять действию помех, будет падать.

Это положение обычно иллюстрируется для случая использования фазовой модуляции (рис. 1.7 и 1.8).

Из рис. 1.7 и рис. 1.8 видно, что расстояние между соседними векторами с ростом J становится все меньше и меньше. Если, скажем, взять $J = 10$, то будем иметь 1024 вектора, различить которые при наличии помех будет достаточно трудно.

В соответствии с теоремой К. Шеннона, пропускная способность канала C определяется выражением

$$C = \Delta F \log_2 \left(1 + \frac{P_c}{P_п}\right) \text{ бит/с}, \qquad (1.3)$$

где P_c — мощность сигнала; $P_п$ — мощность помехи.

При $\frac{P_c}{P_п} = \infty$, т.е. при отсутствии помех пропускная способность стремится к бесконечности. В формуле (1.3) никак не учитывается количество уровней в сигнале. Эта формула устанавливает теоретический предел скорости передачи информации, при которой возможна передача информации со сколь угодно малой вероятностью ошибки. Но ответа на вопрос о том, как это сделать, формула (1.3) не дает.

Одной из задач, стоящих перед разработчиком системы передачи данных, является выбор таких сигналов (или их синтез) для передачи по линии (каналу), которые позволили бы обеспечить передачу информации с максимальной скоростью при заданной вероятности неправильного приема и известной статистике помех. Часто в качестве таких сигналов используются сигналы с квадратурной амплитудной модуляцией (КАМ) — сочетание фазовой и амплитудной модуляции. На рис. 1.9 представлен в качестве примера один из вариантов амплитудно-фазовой модуляции. Здесь один единичный элемент «несет» на себе 4 бита.

1.3. Физическая среда передачи данных

Общие характеристики физической среды. Ниже перечислены характеристики, не зависящие от физической природы среды. Это:

- полоса пропускания;
- пропускная способность;
- задержка;
- затухание (ослабление сигнала);
- помехоустойчивость;
- достоверность передачи;
- стоимость;
- простота прокладки;
- сложность в обслуживании.

Полоса пропускания определяется в герцах и определяет возмож-

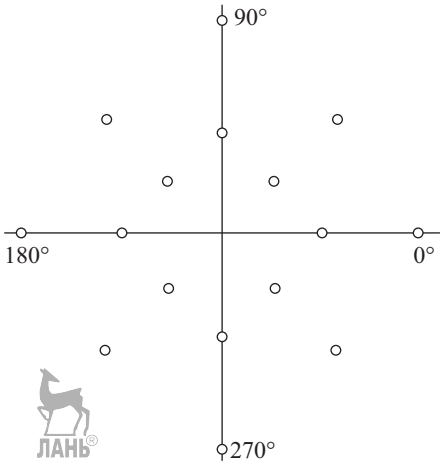


Рис. 1.9. Квадратурная амплитудная модуляция (КАМ-16)

ную скорость передачи информации в данной среде или пропускную способность среды, выраженную в бит/с.

Время распространения сигнала (Δt) в любой среде (задержка) отлично от нуля и зависит от скорости распространения сигнала (V) и расстояния (L), которое проходит сигнал, т.е.

$$\Delta t = \frac{L}{V}.$$

Затухание (α) характеризует потерю мощности передаваемого сигнала в процессе его распространения в среде и определяется в децибелах (дБ).

Представим среду как некий четырехполюсник, мощность сигнала на входе которого $P_{\text{вх}}$, а на выходе $P_{\text{вых}}$. Тогда затухание определяется выражением:

$$\alpha = 10 \lg \frac{P_{\text{вых}}}{P_{\text{вх}}}, \text{ дБ}.$$

Для характеристики среды используется значение километрического затухания, которое определяется в дБ/км.

Помехоустойчивость характеризует степень подверженности среды помехам, возможность их проникновения в среду.

Достоверность (или верность) передачи определяется через коэффициент ошибок

$$k_{\text{ош}} = \frac{n_{\text{ош}}}{n_{\text{пер}}},$$

где $n_{\text{ош}}$ — количество неправильно принятых элементов; $n_{\text{пер}}$ — количество переданных за время испытаний элементов; $k_{\text{ош}}$ служит оцен-

кой для вероятности неправильного приема ($p_{ош}$)

$$p_{ош} = \lim_{n_{пер} \to \infty} \frac{n_{ош}}{n_{пер}}.$$

Остальные характеристики (стоимость, простота организации связи и сложность в обслуживании) вряд ли нуждаются в особых пояснениях.

Физическая среда передачи данных (medium) может представлять собой кабель, земную атмосферу или космическое пространство. В нашем кратком обзоре мы рассмотрим особенности и характеристики таких сред как витая пара, волоконно-оптические линии и радиоканалы наземной и спутниковой связи.

Витая пара. Витой парой (twisted pair) называется скрученная пара проводов. Скручивание проводов позволяет снизить влияние как внешних, так и взаимных помех на полезные сигналы, передаваемые по кабелю. Наличие «взаимных» помех обусловлено тем, что над одной оболочкой располагается множество витых пар, каждая из которых влияет на соседние. Различают влияние на ближний конец (NEXT) и влияние на дальний конец (FEXT).

Показатель NEXT, выраженный в децибелах, равен

$$\text{NEXT} = 10 \lg \frac{P_с}{P_н},$$

где $P_с$ — мощность сигнала на выходе передатчика; $P_н$ — мощность наведенного сигнала.

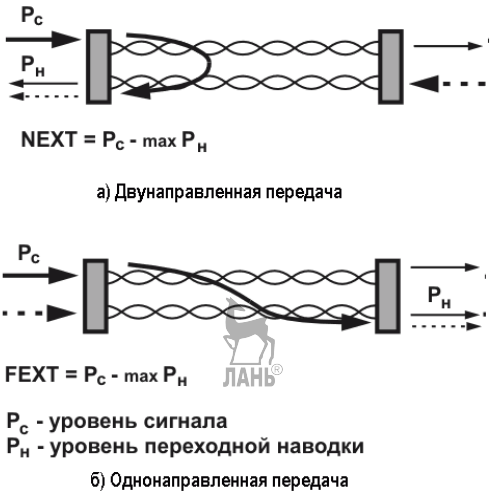


Рис. 1.10. Переходное затухание на ближний (NEXT) и дальний конец (FEXT)

Чем меньше значение NEXT, тем лучше кабель. Перекрестные наводки на дальнем конце позволяют оценить устойчивость кабеля к наводкам для случая, когда передатчик и приемник подключены к разным концам кабеля. Чем меньше уровень помех при заданном уровне сигнала, тем с большей скоростью возможна передача информации.

Кабели на основе витой пары являются симметричными кабелями. Этот кабель может быть как экранированным (Shielded Twisted Pair, STP), так и неэкранированным (Unshielded Twisted Pair, UTP). Экран в первом случае обеспечивает электромагнитную изоляцию, уменьшая влияние внешних помех на полезный сигнал. В качестве такого электромагнитного экрана чаще всего применяется проводящая медная оплетка. Экранированные кабели применяются исключительно при прокладке магистральных линий и в производственных помещениях с высоким уровнем помех. Кабели более высокой категории имеют больше витков на единицу длины. Чем выше категория, тем большая скорость передачи данных может быть обеспечена. Стоимость с ростом категорий также растет. Краткая характеристика UTP кабелей категорий 1–7 приведена ниже [2].

Кабели категории 1 применяются там, где требования к скорости передачи минимальны. Обычно это кабель для цифровой и аналоговой передачи голоса и низкоскоростной (до 20 Кбит/с) передачи данных. До 1983 года это был основной тип кабеля для телефонной разводки.

Кабели категории 2 были впервые применены фирмой IBM при построении собственной кабельной системы. Главное требование к кабелям этой категории — способность передавать сигналы со спектром до 1 МГц.

Кабели категории 3 были стандартизованы в 1991 году. Стандарт EIA-568 определил электрические характеристики кабелей для частот в диапазоне до 16 МГц. Кабели категории 3, предназначенные как для передачи данных, так и для передачи голоса, составляют сейчас основу многих кабельных систем зданий.

Кабели категории 4 представляют собой несколько улучшенный вариант кабелей категории 3. Кабели категории 4 обязаны выдерживать тесты по частоте передачи сигнала 20 МГц и обеспечивать повышенную помехоустойчивость и низкие потери сигнала. На практике используются редко.

Кабели категории 5 были специально разработаны для поддержки высокоскоростных протоколов. Их характеристики по частоте определяются в диапазоне до 100 МГц. Большинство высокоскоростных технологий (FDDI, Fast Ethernet, ATM и Gigabit Ethernet) и ориентировано на использование витой пары категории 5. Кабель категории 5

пришел на замену кабелю категории 3, и сегодня все новые кабельные системы крупных зданий строятся именно на этом типе кабеля (в сочетании с волоконно-оптическим).

Особое место занимают кабели категорий 6 и 7, которые промышленность начала выпускать сравнительно недавно. Для кабеля категории 6 характеристики определяются до частоты 250 МГц, а для кабелей категории 7 — до 600 МГц. Кабели категории 7 обязательно экранируются, причем как каждая пара, так и весь кабель в целом. Кабель категории 6 может быть как экранированным, так и неэкранированным. Основное назначение этих кабелей — поддержка высокоскоростных протоколов на отрезках кабеля большей длины, чем кабель UTP категории 5.

Все кабели UTP независимо от их категории выпускаются в 4-парном исполнении. Каждая из четырех пар кабеля имеет определенный цвет и шаг скрутки. Обычно две пары предназначены для передачи данных, две — для передачи голоса.

Экранированная витая пара хорошо защищает передаваемые сигналы от внешних помех, а также меньше излучает электромагнитные колебания вовне, что, в свою очередь, защищает пользователей сетей от вредного для здоровья излучения. Наличие заземляемого экрана удорожает кабель и усложняет его прокладку.

Экран выполняется либо плетеным из медной проволоки (хорошо защищает от низкочастотных наводок), либо из токопроводящей фольги (пленки), которая блокирует высокочастотное электромагнитное излучение. Также на практике часто используют двойные экраны (HIGHT Screen), в которых используются оба способа. Эффект от применения экрана заключается в уменьшении внешних наводок на экранированную пару (или несколько пар), и снижении уровня их электромагнитного излучения «наружу». Но общий экран вызывает рост NEXT (перекрестных наводок) из-за отражения от экрана, на 10–20%. Далее, экранирование увеличивает затухание в кабеле вследствие добавочной емкости между экраном и витыми парами. Кроме того, монтаж экранированной системы значительно более сложен (дорог), требует хорошего подбора всех элементов. А самые незначительные ошибки способны ухудшить, а не улучшить параметры линии.

Основным стандартом, определяющим параметры экранированной витой пары для применения внутри зданий, является фирменный стандарт IBM. В этом стандарте кабели делятся не на категории, а на типы от 1 до 9 включительно.

Рассмотрим для примера кабель типа 1 стандарта IBM. Он состоит из двух пар скрученных проводов, экранированных проводящей

оплеткой, которая заземляется. Электрические параметры кабеля типа 1 примерно соответствуют параметрам кабеля UTP категории 5. Однако волновое сопротивление кабеля типа 1, равное 150 Ом, значительно выше волнового сопротивления кабеля UTP категории 5 (100 Ом), поэтому невозможно «улучшение» кабельной проводки сети путем простой замены неэкранированной пары экранированной парой типа 1. Передатчики, рассчитанные на работу с кабелем, имеющим волновое сопротивление 100 Ом, будут плохо работать на волновое сопротивление 150 Ом.

Волоконно-оптические линии связи (ВОЛС). Волоконно-оптический кабель состоит из тонких гибких стеклянных волокон, по которым распространяются световые сигналы. Каждый световод состоит из центрального проводника света (сердцевины) и внешней оболочки, обладающей меньшим, чем сердцевина показателем преломления. Распространяясь по сердцевине, лучи света не выходят за ее пределы, отражаясь от покрывающего слоя оболочки. Лучи, входящие под разными углами в оптоволокно, называют *модами*. Волокно, по которому распространяется несколько мод, называется многомодовым. По одномодовому волокну распространяется только один луч. Технические характеристики многомодовых кабелей хуже, чем у одномодовых. Многомодовые кабели применяют в основном для передачи данных на скоростях не более 1 Гбит/с на небольшие расстояния, одномодовые — для передачи на сверхвысоких скоростях на дальние расстояния.

Для связи используется диапазон от 850 нм до 1625 нм. В этом диапазоне существуют окна прозрачности — поддиапазоны, в которых затухание уменьшается (рис. 1.11). Это 0,85 мкм, 1,3 мкм, 1,55 мкм. В соответствии с этими окнами выпускаются и излучатели.

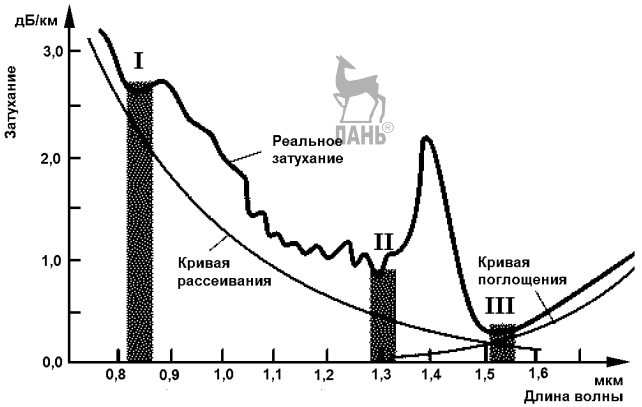


Рис. 1.11. Зависимость затухания от длины волны

На данное время все высокоскоростные системы оптической связи работают в одном из трех диапазонов:

- С-полоса (1530...1565 нм);
- L-полоса (1565...1620 нм);
- окно прозрачности вблизи 1,3 мкм.

Минимальное ослабление сигнала имеет место в диапазоне С (порядка 0,15 дБ/км).

ВОЛС в используемом диапазоне частот отличаются чрезвычайно широкой полосой пропускания (порядка 20 ТГц). Это позволяет разместить в одном волокне 250 миллионов стандартных цифровых каналов со скоростью 64 Кбит/с.

Так, в третьем окне прозрачности затухание порядка 0,2 дБ/км. При допустимом затухании 20 дБ максимальное расстояние между усилителями составляет порядка 100 км и более. Причиной затухания являются:

- рассеяние энергии из-за микроскопических неоднородностей в волокне;
- поглощение, обусловленное преобразованием энергии света в тепловую энергию из-за микровкраплений;
- потери на стыках волокон вследствие неточной центровки, качества стыков и др.;
- потери на изгибах, вызывающих выход излучения за пределы сердцевины и поглощения в оболочке.

Все перечисленные выше причины носят технологический характер. Технология изготовления волокна постоянно совершенствуется, что приводит к снижению затухания. Достаточно напомнить тот факт, что нить изготовленная в 1970 г., имела затухание 20 дБ/км.

Волоконно-оптический кабель характеризуется низким уровнем шумов, что позволяет обеспечить передачу информации с высокой скоростью (вспомните формулу К. Шеннона, представленную в разделе 1.2). Волокно невосприимчиво к электромагнитным помехам со стороны окружающих медных кабелей и электрического оборудования. В многоволоконных кабелях не существует проблем NEXT, FEXT. Волоконно-оптический кабель имеет меньший вес по сравнению с медными кабелями (диаметр волокна порядка 100 микрон). Они могут применяться в авиации, приборостроении. Основой оптоволокна является кварц (SiO_2), самый распространенный в природе материал и менее дорогой, чем медь.

Волоконно-оптический кабель обеспечивает высокую безопасность от несанкционированного доступа, так как перехват информации возможен только с использованием способов с разрушением среды, целостность которой постоянно контролируется. И еще одно

достоинство — длительный срок эксплуатации (порядка 25 лет).

Отметим также некоторые проблемы, характерные для современного этапа внедрения волоконно-оптической связи, когда оптика и электроника используется совместно:

- электроника отстает от оптики по частотам;
- оконечное оборудование с электрооптическими и оптоэлектрическими преобразователями очень дорогое;
- для мощных электрооптических преобразователей требуется охлаждение.

Методы устранения перечисленных выше недостатков очевидны — это полный переход на оптику. Следует также иметь в виду следующие сложности, связанные с использованием ВОЛС: для монтажа оптоволоконных линий требуется прецизионное оборудование, восстановление работоспособности при отказах на магистрали требует бóльших затрат по сравнению с кабельными и радиорелейными линиями связи.

Радиоканалы образуются с помощью передатчика и приемника радиоволн [2, 3]. Существует большое разнообразие типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью связи. Диапазоны широковещательного радио (длинных, средних и коротких волн), называемые также AM-диапазонами или диапазонами амплитудной модуляции (Amplitude Modulation, AM), обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, использующие диапазоны очень высоких частот (Very High Frequency, VHF), для которых применяется частотная модуляция (Frequency Modulation, FM). Для передачи данных также используются диапазоны ультравысоких частот (Ultra High Frequency, UHF), называемые еще диапазонами микроволн (свыше 300 МГц). При частоте свыше 30 МГц сигналы уже не отражаются ионосферой Земли, и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником. Поэтому указанные частоты используются в спутниковых или радиорелейных каналах либо в таких локальных или мобильных сетях, в которых это условие выполняется.

Беспроводные каналы используются чаще всего в тех случаях, когда кабельные линии связи применить нельзя, например, при прохождении канала через малонаселенную местность или же для связи с мобильными пользователями сети. Обеспечение мобильности затронуло в первую очередь телефонные сети, компьютерные сети в этом отношении пока отстают. Тем не менее, построение компьютерных сетей на основе беспроводных технологий, например Radio Ethernet LTE, считается сегодня одним из самых перспективных на-

правлений телекоммуникаций. Высокоскоростная передача данных на основе беспроводной среды рассматривается в главе 7.

1.4. Методы преобразования сигналов

Задача по доставке информации из одного пункта в другой решается с использованием телекоммуникационных систем — средств, обеспечивающих электрическую связь (электросвязь) определенного типа.

В приведенном определении есть ключевые слова «связь» и «электросвязь» определенного типа. Что же это такое?

Связь (communication) — обмен информацией или пересылка информации с помощью средств, функционирующих в соответствии с согласованными правилами (называемыми в конкретных условиях протоколами).

Международная конвенция по электросвязи (Найроби, 1982 г.) определила «электросвязь» как «...передачу, получение и прием знаков, сигналов, письменного текста, изображения и звуков или сообщений любого рода по проводной, радио и оптической или другим электромагнитным системам...».

Обобщенная структурная схема системы электросвязи (телекоммуникационной системы) приведена на рис. 1.12.

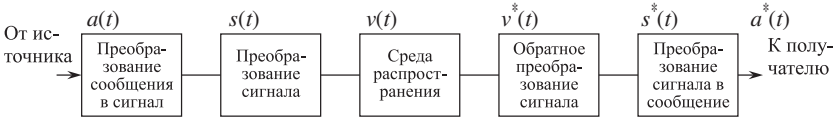


Рис. 1.12. Система электросвязи

Источник сообщения формирует сообщение $a(t)$, которое с помощью специальных устройств преобразуется в электрический сигнал $s(t)$. При передаче речи такое преобразование выполняет микрофон, при передаче изображения — передающая телевизионная трубка, при передаче данных — компьютер.

Чтобы передать сигнал в системе электросвязи, нужно воспользоваться каким-либо переносчиком. В качестве переносчика естественно использовать те материальные объекты, которые имеют свойство перемещаться в пространстве, например, электромагнитное поле в проводах (проводная связь), в открытом пространстве (радиосвязь), световой луч (оптическая связь).

Таким образом, в пункте передачи (рис. 1.12) первичный сигнал $s(t)$ необходимо преобразовать в сигнал $v(t)$, удобный для его передачи по соответствующей среде распространения. В пункте приема выполняется обратное преобразование.

Существуют различные варианты преобразования сообщения в сигнал, суть которых в конечном счете сводится к задаче согласования сигнала со средой распространения. На рис. 1.12 изображен случай, когда имеется один источник и один получатель. В ряде случаев среда используется для передачи информации от нескольких источников. В последнем случае появляется необходимость обеспечить так называемый множественный доступ к среде.

Линейное кодирование. Так называется преобразование последовательности прямоугольных импульсов (цифровых сигналов) в цифровой сигнал $v(t)$, пригодный для передачи по линии. При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей [2]:

- минимизировать ширину спектра сигнала, полученного в результате кодирования;
- обеспечивать синхронизацию между передатчиком и приемником;
- обеспечивать устойчивость к шумам;
- обнаруживать и по возможности исправлять битовые ошибки;
- минимизировать мощность передатчика.

Более узкий *спектр сигнала* позволяет на одной и той же линии (с одной и той же полосой пропускания) добиваться более высокой скорости передачи данных. Спектр сигнала в общем случае зависит как от способа кодирования, так и от тактовой частоты передатчика.

Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времен считывать новую порцию информации с линии связи. При передаче дискретных сообщений время всегда разбивается на такты одинаковой длительности, а приемник старается считать новый сигнал в середине каждого такта, то есть синхронизировать свои действия с передатчиком.

На рис. 1.13 приведены варианты линейного (дискретного [2]) кодирования данных.

На рис. 1.13,а представлено линейное кодирование без возвращения к нулю (Non Return to Zero, NRZ). Из названия следует, что при передаче последовательности единичных элементов сигнал не возвращается к нулю в течение такта.

Достоинства метода NRZ [2]:

- простота реализации;
- метод обладает хорошей распознаваемостью появления ошибки (благодаря наличию двух резко отличающихся потенциалов);
- основная гармоника f_0 имеет достаточно низкую частоту, что обеспечивает более узкий спектр.

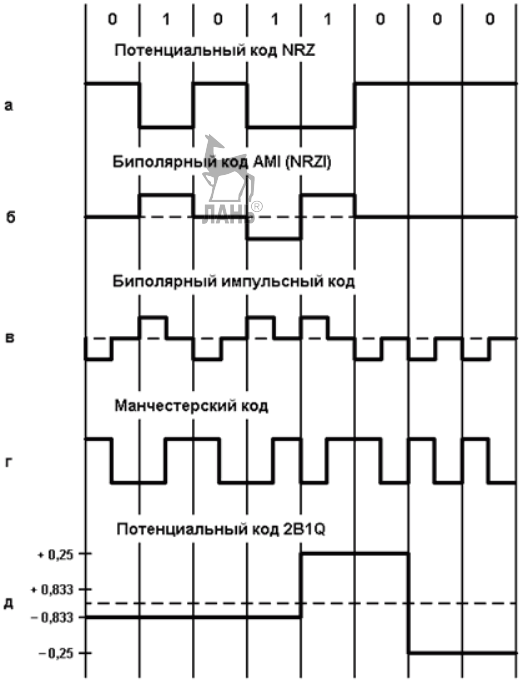


Рис. 1.13. Способы линейного кодирования

Недостатки метода NRZ:

– метод не обладает свойством самосинхронизации. Даже при наличии высокоточного тактового генератора приемник может ошибиться с выбором момента съема данных, так как частоты двух генераторов никогда не бывают полностью идентичными. Поэтому при высоких скоростях обмена данными и длинных последовательностях единиц или нулей небольшое рассогласование тактовых частот может привести к ошибке в целый такт и, соответственно, считыванию некорректного значения бита;

– вторым серьезным недостатком метода NRZ является наличие низкочастотной составляющей, частота которой приближается к нулю при передаче длинных последовательностей нулей. Из-за этого многие линии связи, не обеспечивающие прямого гальванического соединения между приемником и источником, этот вид кодирования не поддерживают. Поэтому в сетях код NRZ в основном используется в виде различных его модификаций, в которых устранены проблемы плохой самосинхронизации и постоянной составляющей.

Одной из модификаций метода NRZ является метод *биполярного кодирования с альтернативной инверсией* (Alternate Mark Inversion, AMI). В этом методе применяются три уровня потенциала — отрицательный, нулевой и положительный (см. рис. 1.13,б). Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

При передаче длинных последовательностей единиц код AMI частично решает проблемы наличия постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, то есть без постоянной составляющей и с основной гармоникой $N/2$ Гц (где N — битовая скорость передачи данных). Длинные же последовательности нулей для кода AMI столь же опасны, как и для кода NRZ — сигнал вырождается в постоянный потенциал нулевой амплитуды.

В целом, для различных комбинаций единичных элементов на линии использование кода AMI приводит к более узкому спектру сигнала, чем для кода NRZ, а значит, и к более высокой скорости передачи информации. Например, при передаче чередующихся единиц и нулей основная гармоника f_0 имеет частоту $N/4$ Гц.

Код AMI предоставляет также некоторые возможности по распознаванию ошибочных сигналов. Так, нарушение строгой очередности в полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса.

В коде AMI используются не два, а три уровня сигнала на линии. Дополнительный уровень требует увеличения мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема битов на линии, что является общим недостатком кодов с несколькими состояниями сигнала по сравнению с кодами, в которых различают только два состояния.

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен на предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется потенциальным кодом с инверсией при единице (Non Return to Zero with ones Inverted, NRZI). Он удобен в тех случаях, когда наличие третьего уровня сигнала весьма нежелательно, например, в оптических кабелях, где устойчиво распознаются только два состояния сигнала — свет и темнота.

Код NRZI хорош тем, что в среднем требует меньше изменений сигнала при передаче произвольной двоичной информации, чем манчестерский код, за счет чего спектр его сигналов уже. Однако код NRZI обладает плохой самосинхронизацией, так как при передаче длинных последовательностей нулей сигнал вообще не меняется (например, при передаче последних 3-х нулей на рис. 1.13,а), и, значит, у приемника исчезает возможность синхронизации с передатчиком на значительное время, что может приводить к ошибкам распознавания данных.

Для улучшения линейных потенциальных кодов, подобных AMI и NRZI, используются два метода. Первый метод основан на добавлении в исходный код избыточных битов, содержащих логические единицы. Очевидно, что в этом случае длинные последовательности нулей прерываются, и код становится самосинхронизирующимся для любых передаваемых данных. Исчезает также постоянная составляющая, а значит, еще более сужается спектр сигнала. Однако этот метод снижает скорость передачи информации, так как избыточные единицы пользовательской информации не несут.

Другой метод основан на предварительном «перемешивании» исходной информации таким образом, чтобы вероятность появления длинных последовательностей единиц и нулей на линии становилась близкой к нулю. Устройства или блоки, выполняющие такую операцию, называются *скремблерами*. При скремблировании используется известный алгоритм, поэтому приемник, получив двоичные данные, передает их на дескремблер, который восстанавливает исходную последовательность битов.

Помимо потенциальных кодов для передачи по линии используются и импульсные коды, в которых данные представлены полным импульсом или же его частью — фронтом. Наиболее простым кодом такого рода является биполярный импульсный код, в котором единица представляется импульсом одной полярности, а ноль — другой (см. рис. 1.13,в). Каждый импульс длится половину такта. Подобный код обладает отличными самосинхронизирующимися свойствами, но постоянная составляющая может присутствовать, например, при передаче длинной последовательности единиц или нулей. Кроме того, спектр у него шире, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода равна N Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

В локальных сетях до недавнего времени самым распространен-

ным был так называемый манчестерский код (см. рис. 1.13,г). Он применяется в технологиях Ethernet и Token Ring.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль — обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется, по крайней мере, один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода у́же, чем у биполярного импульсного. У него также нет постоянной составляющей, к тому же основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и нулей) — $N/2$ Гц, как и у кодов AMI и NRZ. В среднем ширина полосы манчестерского кода в полтора раза у́же, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения $3N/4$. Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском — два.

Для улучшения кодов типа AMI, NRZI используют логические линейные коды и скремблирование.

Избыточные линейные коды основаны на разбиении исходной последовательности битов на порции, которые часто называют символами [2]. Затем каждый исходный символ заменяется новым с большим количеством битов, чем исходный. В качестве примера приведем код 4B/5B, используемый в технологиях FDDI и Fast Ethernet. Здесь исходные символы длиной 4 бит заменяются символами длиной 5 бит. Так как результирующие символы содержат избыточные элементы, то общее количество возможных комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 комбинации, в то время как исходные символы — только 16. Поэтому в результирующем коде можно отобрать 16 таких комбинаций, которые не содержат большого количества нулей, а остальные считать *запрещенными кодовыми комбинациями*. Помимо устранения постоянной составляющей и придания коду свойства самосинхронизации, логические коды позволяют приемнику обнаруживать появление ошибок. Если приемник принимает запрещенную кодовую комбинацию, значит, на линии произошло искажение сигнала.

Далее получившийся код 4B/5B передается по линии путем преобразования с помощью какого-либо из методов потенциального кодирования, чувствительного только к длинным последовательностям нулей. Таким кодом является, например, код NRZI.

Модуляция. Использование различных методов модуляции направлено на такое преобразование исходного сигнала, которое позволит ему «вписаться» в канал. Рассмотрим сигнал, спектр которого расположен в диапазоне 0...200 Гц. Через стандартный телефонный канал (0,3...3,4 кГц) такой сигнал «не пройдет». Необходимо исходный спектр перенести в диапазон 0,3...3,4 кГц. Для этого используем несущую $U_н(t)$

$$U_н(t) = E_н \sin(\omega_н t + \Delta\varphi),$$

где $E_н$ — амплитуда несущей, $\omega_н$ — частота несущей, $\Delta\varphi$ — начальная фаза несущей.

Воздействуя сигналом $U_c(t)$ на различные параметры несущей, получим амплитудную (АМ), частотную (ЧМ) и фазовую модуляцию (ФМ) (см. рис. 1.14). Особенности этих видов модуляции изложены в ряде источников ([1, 2] и других).

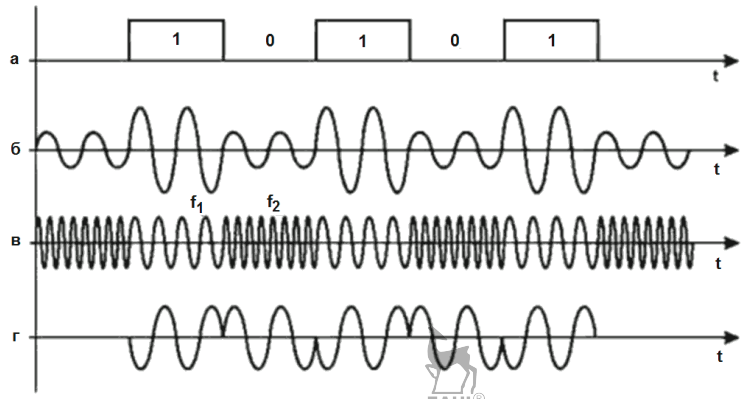


Рис. 1.14. Амплитудная, частотная и фазовая модуляция

При использовании в качестве модулирующего сигнала последовательности прямоугольных импульсов иногда вместо термина «модуляция» используется термин «манипуляция».

Часто в системах высокоскоростной передачи данных используются комбинированные методы модуляции, а именно квадратурная амплитудная модуляция (QAM) — сочетание амплитудной и фазовой модуляции [1].

Сегодня все большую популярность приобретает технология OFDM.

OFDM (Ortogonal Frequency Division Multiplexing) является многочастотным видом модуляции, что подразумевает использование множества генераторов ортогональных несущих частот для формирования сигнала.

В [4] можно встретить следующую расшифровку аббревиатуры OFDM, которая используется в русскоязычных работах: мультиплексирование с ортогональным частотным разделением каналов; мультиплексирование по ортогональным несущим частотам; ортогональное частотное мультиплексирование с разделением частот; модуляция с ортогональным разделением по частотам. Нетрудно заметить, что фактически ни одно из определений не дает дословного перевода всех составляющих аббревиатуры, поскольку сама аббревиатура не может вместить всех особенностей метода модуляции, сущность которого заключается в использовании для передачи исходной цифровой последовательности множества ортогональных несущих. Эта последовательность сначала поступает на демультиплексор, где осуществляется преобразование последовательного кода в параллельный код. Так мы получаем N последовательностей, которые используются для модуляции N ортогональных несущих. С выхода модуляторов модулированный сигнал поступает на сумматор и далее в канал.

На рис. 1.15 показано формирование OFDM сигнала на основе четырех несущих.

Процесс преобразования последовательного кода в демультиплексоре (DMUX) в параллельный приведем для случая, когда $N = 4$ (рис. 1.16).

Протокольная единица, передаваемая с помощью одной несущей, называется символом. Продолжительность символа, используемого для модуляции несущей в N раз больше, чем длительность единичного элемента в исходной последовательности (τ_0). Это делает OFDM сигнал более устойчивым к межсимвольной интерференции и к многолучевому распространению в радиосреде.

Другой особенностью OFDM является, как уже было упомянуто выше, использование ортогональных несущих. Несущие $S_i(t)$ и $S_j(t)$ являются ортогональными, если

$$\int_0^{N\tau_0} S_i(t)S_j(t)\,dt = 0.$$

Так, ортогональными являются несущие, взятые с шагом, равным скорости манипуляции.

При ортогональном методе демодуляции взаимные помехи от соседних несущих группового тракта будут равны нулю, несмотря на то, что их соседние боковые полосы взаимно перекрываются.

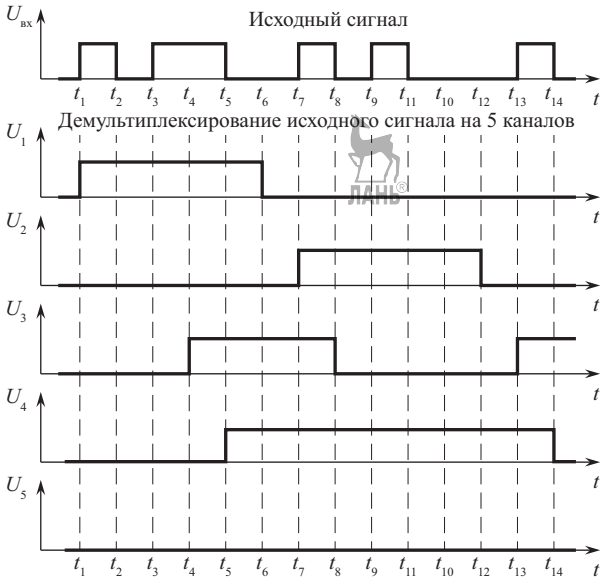


Рис. 1.15. Формирование OFDM сигнала

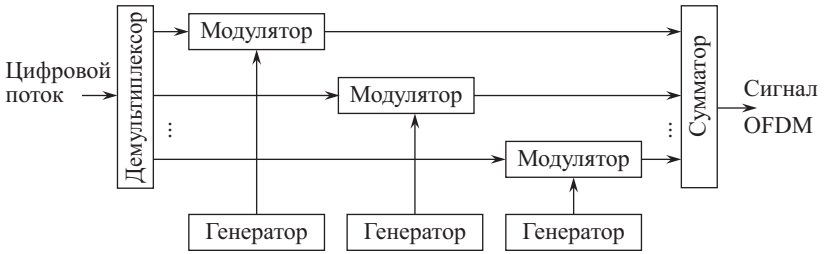


Рис. 1.16. Преобразование последовательного кода в параллельный

Перечислим достоинства OFDM:

1) высокая эффективность использования спектра (спектры несущих OFDM сигнала частично перекрываются, а несущие ортогональны друг другу);
2) простая аппаратурная реализация;
3) хорошая устойчивость к межсимвольной интерференции и многолучевому распространению сигналов;
4) возможность использовании на каждый из N несущих своего вида модуляции (например, на одной несущей 8QAM, на другой 16QAM и т.п.). Это позволяет адаптироваться к уровню помех в различных частотных поддиапазонах, изменяя скорость передачи информации.

К недостаткам следует отнести:

1) необходимость высокой точности синхронизации по частоте и времени;
2) чувствительность к эффекту Доплера (сдвигу частот, зависящему от скорости передвижения приемника), ограничивающему применение OFDM в мобильных системах при высокой скорости передвижения;
3) снижение спектральной эффективности за счет использования так называемого «циклического префикса» — защитного интервала между протокольными единицами.

Несмотря на эти недостатки, различные модификации OFDM широко используются в современных высокоскоростных системах связи и, в частности, WiMAX и LTE.

1.5. Методы множественного доступа к среде

Метод доступа к среде — это алгоритм, согласно которому узлы сети (рабочие станции) получают доступ к среде передачи данных.

Методы CSMA/CD и CSMA/CA. Метод CSMA/CD (Carrier Sense Multiple Access with Collision Detection) — это множественный доступ с контролем несущей и обнаружением коллизий.

Алгоритм работы сетевого адаптера рабочей станции при использовании метода CSMA/CD заключается в следующем:

1) рабочая станция «прослушивает» канал, проверяя, не передает ли кто-либо данные;
2) если «слышит» чью-либо передачу, ожидает ее окончания;
3) если решает, что канал свободен, начинает передачу пакета;
4) при обнаружении коллизии во время передачи прекращает передачу;
5) через случайный промежуток времени все повторяется (т.е. осуществляется переход к п. 1).

Отметим, что коллизии при данном методе возникают, когда канал, будучи свободным, пытаются одновременно занять несколько рабочих станций. При большом количестве рабочих станций и высокой нагрузке число столкновений (коллизий) растет, а пропускная способность канала падает.

Достоинством метода является простая реализация, по причине которой метод был предложен для использования в технологии Ethernet в 1976 г.

Метод CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) — метод множественного доступа с контролем несущей и предотвращением столкновений. При этом варианте множественного доступа перед передачей данных рабочая станция посылает в сеть короткий пакет, сообщая остальным рабочим станциям о своем наме-

рении начать передачу. Таким образом, другие рабочие станции ставятся в известность о готовящейся передаче, что позволяет избежать коллизий. Наличие уведомлений, несомненно, увеличивает общую нагрузку на среду, однако отсутствие коллизий делает применение метода CSMA/CA в ряде случаев более предпочтительным, чем метода CSMA/CD. В частности, такой метод применяется в беспроводных сетях.

Существует также целый ряд других методов доступа, исключающих коллизии. Это мультиплексирование по частоте, длине световой волны, времени и другие.

Частотное разделение каналов (Frequency Division Multiplexing, FDM). Пусть нам предоставлен в распоряжение частотный диапазон $\Delta\omega$. В то же время источник вырабатывает сигналы, занимающие спектр частот в диапазоне $\Delta\Omega$. Пусть также имеет место соотношение $\Delta\omega \gg \Delta\Omega$. Использовать диапазон $\Delta\omega$ для передачи сигналов от одного источника было бы неразумно (неэкономично). Очевидно, что в диапазоне $\Delta\omega$ можно было бы организовать приблизительно $N = \frac{\Delta\omega}{\Delta\Omega}$ каналов для передачи информации от N источников.

Частотное разделение (уплотнение) каналов (ЧРК) предусматривает выделение каждому источнику сигналов фиксированного, строго определенного места в общем частотном диапазоне.

Для этого выделенный для организации связи частотный диапазон $\Delta\omega$, предназначенный для связи, разбивается на N (по количеству каналов) частотных поддиапазонов (рис. 1.17).

В каждый такой поддиапазон $\Delta\omega_i$ «помещают» спектр соответствующего канального сигнала $S_i(\omega)$. Такой способ основан на том, что спектр реального сигнала практически ограничен определенным интервалом частот. Считается, что вне этого интервала спектральные составляющие сигнала отсутствуют. С помощью канальных передатчиков спектры сообщений $S_i(\Omega)$ не меняя своей структуры, преобразуются в канальные сигналы $S_i(\omega)$. Задача канальных передатчиков — распределить исходные спектры сообщения $S_i(\Omega)$ по частоте, выстроив их друг за другом. Таким образом, при частотном объединении каналов мы должны «поместить» каждый канальный спектр $S_i(\omega)$ в соответствующий отдельный поддиапазон $\Delta\omega_i$.

Способ построения каналообразующей аппаратуры с ЧРК показан на рис. 1.18. На передающей стороне каждый из канальных передатчиков (КП) имеет собственную несущую частоту ω_i. Частоты подобраны таким образом, чтобы сигналы на выходах передатчиков были разнесены по спектру (в соответствии с рис. 1.17). Сумма всех канальных сигналов образует групповой сигнал, который поступает в линию.

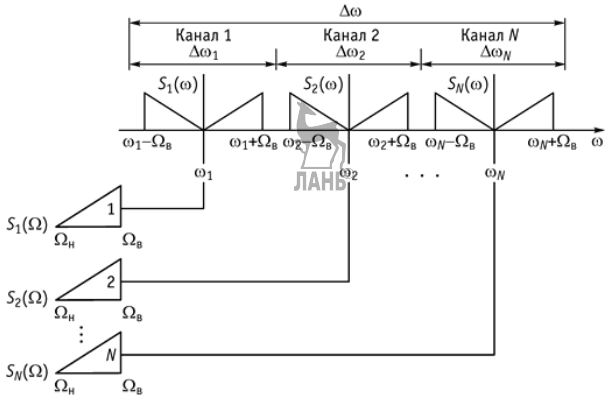


Рис. 1.17. Перенос спектра исходных сигналов при ЧРК

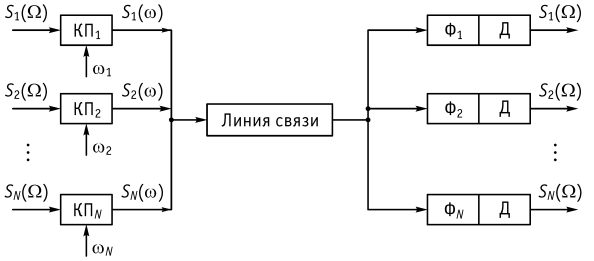


Рис. 1.18. Построение многоканальной системы с ЧРК

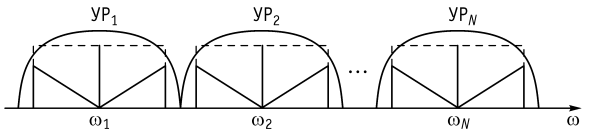


Рис. 1.19. Разделение сигналов на приемной стороне

На приемной стороне разделение происходит за счет фильтрации отдельных канальных спектров (рис. 1.19). Канальные фильтры $\Phi_1, \Phi_2, ..., \Phi_N$ — своеобразное спектральное решето. Каждый фильтр пропустит только «свой» спектр и задержит остальные. Выделенный фильтром сигнал преобразуется в исходный с помощью детектора *Д* как в обычной одноканальной системе.

Для того, чтобы при многоканальной передаче не возникли помехи между каналами, необходимо ширину частотных поддиапазонов $\Delta\omega_i$ взять несколько большей ширины канальных спектров $S_i(\omega)$ группового сигнала, т.е. они не должны располагаться друг за другом «впритык». Между ними должен быть частотный промежуток —

защитный интервал. В противном случае при разделении сигналов составляющие соседних канальных спектров могут просочиться друг к другу и вызвать искажения. Причиной является несовершенство характеристик устройств разделения (УР). На рис. 1.19 штриховыми линиями показаны требуемые (идеальные) характеристики фильтров, а сплошными — реальные.

В качестве варианта частотного уплотнения можно рассматривать спектральное волновое уплотнение, используемое при передаче информации по волоконно-оптическим линиям связи.

Спектральное волновое уплотнение каналов (Wavelength-Division Multiplexing, WDM) или мультиплексирование по длине волны — технология, которая позволяет организовать несколько информационных каналов по одному оптическому волокну. Одним из примеров является передача запроса на получение информации от потребителя услуг на волне 1310 нм и получение этой информации на волне 1550 нм.

Современные WDM-системы на основе стандартного частотного плана (ITU-T Rec. G. 692) делятся на три группы:

1) грубое разделение (Coarce WDM, CWDM) — системы с частотным разносом каналов более 2500 ГГц;
2) плотное разделение (Dence WDM, DWDM) — системы с разносом 100 ГГц и 50 ГГц;
3) высокоплотное разделение (High Dence WDM, HDWDM) — системы с разносом каналов 25 ГГц.

С успехами в области технологий WDM связано создание полностью оптических сетей. В таких сетях все операции осуществляются без преобразования сигналов из оптической формы в электрическую, что позволяет существенно удешевить сеть.

Множественный доступ OFDMA является многопользовательской версией OFDM. При OFDMA пользователям приписываются свои наборы несущих, что позволяет обеспечить одновременную передачу данных для нескольких абонентов.

Кодовое разделение (уплотнение) [2]. Технология CDMA (Code Division Multiple Access — множественный доступ с кодовым разделением) отличается тем, что все абоненты используют *одновременно одну и ту же полосу частот, отсутствует и разделение по времени*. Возникает вопрос: как опознать на приемном конце сигналы разных абонентов? Это допускает использование для каждого канала (источника) своей уникальной кодовой последовательности, накладываемой на передаваемый сигнал (рис. 1.20). Чтобы извлечь сигнал на приемном конце, надо знать используемую для него кодовую последовательность.

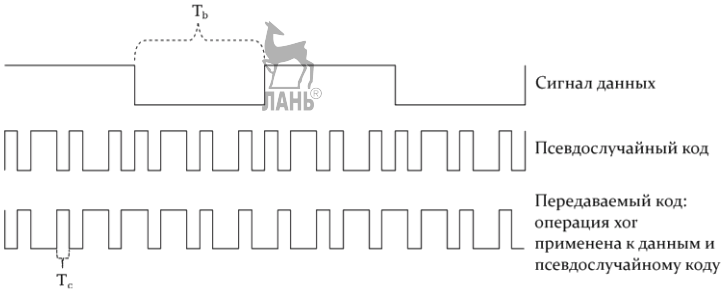


Рис. 1.20. Формирование сигнала на передаче

Технология CDMA используется в сотовых системах связи и имеет следующие достоинства:

- емкость базовых станций увеличивается в 45 раз по сравнению с GSM;
- отсутствие частотного планирования благодаря использованию тех же самых частот в смежных секторах каждой соты;
- улучшенная защищенность передаваемых данных (для выделения сигналов на приемном конце надо знать вид кодовой последовательности);
- улучшенные характеристики покрытия, позволяющие использовать меньшее количество сот;
- большее время работы батарей до разрядки;
- возможность выделения требуемой полосы частот по потребности.

Недостатки:

- сложное системное планирование (размещение базовых станций на местности);
- необходимость многоступенчатого управления мощностью передачи мобильных станций;
- жесткие требования в синхронизации в сетях CDMA.

Временно́й способ разделения каналов (Time Division Multiplexing, TDM). Пусть, как и ранее, в нашем распоряжении имеется частотный диапазон $\Delta\omega$, а источник «вырабатывает» сигналы, занимающие полосу частот $\Delta\Omega$. Пусть также имеет место соотношение $\Delta\omega \gg \Delta\Omega$. Предположим, что сигналы, поступающие от источников, представляют собой последовательность прямоугольных импульсов, длительность которых τ_0. Так как $\Delta\omega \gg \Delta\Omega$, то, как следует из раздела 1.2, в частотном диапазоне $\Delta\omega$ можно передавать импульсы $\tau_k \ll \tau_0$. Напомним, что чем короче импульсы, тем шире полоса частот, требуемая для его передачи без искажений. И следовательно, более широкая полоса частот позволяет передавать более короткие импульсы. На ин-

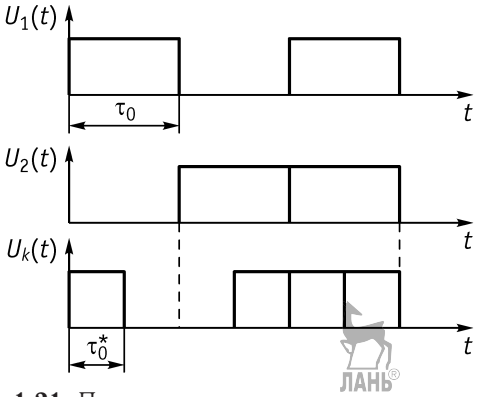


Рис. 1.21. Принцип временного разделения каналов

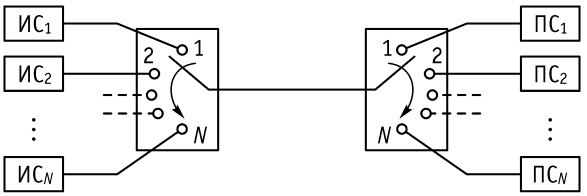


Рис. 1.22. Система передачи с временным разделением каналов

тервале τ_0 можно разместить примерно N импульсов длительностью τ_k, т.е. $N = \frac{\tau_0}{\tau_k} = \frac{\Delta\omega}{\Delta\Omega}$.

Рассмотрим случай, когда $\Delta\omega = 2\Delta\Omega$. На рис. 1.21 показаны сигналы, поступающие от двух источников $U_1(t)$ и $U_2(t)$, и сигналы в канале связи $\tau_0^* = \frac{\tau_0}{2}$.

Таким образом, на интервале τ_0 мы передали информацию от двух источников, т.е. мы «уплотнили» этот интервал!

На рис. 1.22 изображена упрощенная схема системы с временным разделением (уплотнением) каналов (ВРК) для случая, когда $\frac{\Delta\omega}{\Delta\Omega} = N$.

Источники и получатели сообщений подключаются к среде передачи (линии) поочередно с помощью двух специальных коммутаторов, работающих согласованно — синхронно и синфазно. Подключение источников (получателей) осуществляется на интервал времени τ_k. После того, как будет подключен N-ый источник (получатель), подключается первый и т.д. Следовательно, системы с ВРК работают циклично и непрерывно.

Временное разделение каналов сегодня используется очень часто, иногда совместно с частотным, как, например, в системах мобильной радиосвязи стандарта GSM.

Временные интервалы или частотные интервалы используются рабочими станциями по мере надобности и могут, как нетрудно догадаться, простаивать, что в условиях нехватки ресурсов является недостатком методов доступа TDM и FDM.

1.6. Сети электросвязи

Сети электросвязи, в конечном счете, создаются для удовлетворения потребностей человека (рис. 1.23)

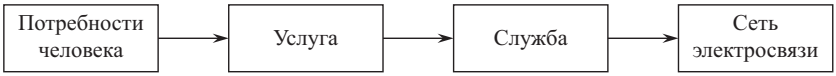


Рис. 1.23. Зачем нужна сеть электросвязи?

Под услугой понимается возможность удовлетворения потребностей человека. Служба — это совокупность организационно-технических мер, которые позволяют организовать услугу. Сеть связи — совокупность технических средств, обеспечивающих поддержку одной или нескольких служб. Сеть связи еще называют телекоммуникационной сетью.

Модель телекоммуникационной системы, предложенная ITU (International Telecommunication Union), изображена на рис. 1.24.

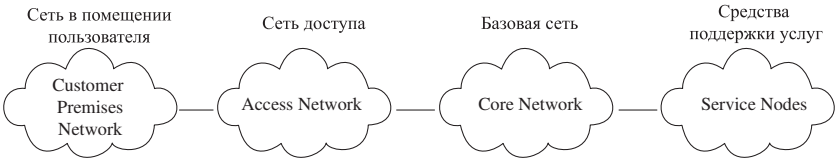


Рис. 1.24. Модель телекоммуникационной системы, предложенная ITU

Отметим, что телекоммуникационную сеть следует рассматривать как некую систему. Слово «система» происходит от греческого $\sigma\upsilon\sigma\tau\eta\mu\alpha$ — целое, составленное из частей, соединение. Система — множество элементов, находящихся в отношениях и связях друг с другом, которое образует определенную целостность, единство [5].

Итак, телекоммуникационная сеть — это система, но не всякая система является сетью. Отсюда понятно, что понятие система более общее, чем сеть.

Сегодня в специальной литературе можно встретить понятие «инфокоммуникационная сеть» (старое название «компьютерная сеть»). Это технологическая система, которая включает в себя кроме средств доставки информации также средства хранения, обработки и поиска информации.

В последнее время появился ряд работ, в которых рассматриваются когнитивные инфокоммуникационные сети [6, 7], программно-

конфигурированные сети. «Когнитивная сеть — сеть с познавательным процессом, который позволяет учитывать текущее состояние сети, а затем планировать, принимать решение и действовать с учетом текущего состояния сети» [7]. С целью знакомства с идеологией когнитивных сетей мы отправим читателя к упомянутым ранее первоисточникам, а программно-конфигурированным сетям посвятим отдельную главу, исходя из того, что первые сегодня еще очень далеки от практической реализации, а ко вторым проявляют интерес многие вендоры и операторы, включая российских.

По территориальной распространенности сети делятся на локальные (Local Area Network, LAN), региональные (Metropolitan Area Network MAN) и глобальные (Wide Area Network, WAN) (рис. 1.25).

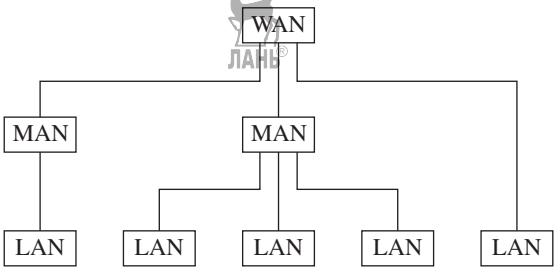


Рис. 1.25. Иерархия сетей электросвязи

Локальной называется сеть, абоненты которой находятся на небольшом (до 10–15 км) расстоянии друг от друга. Локальные вычислительные сети (ЛВС) объединяют абонентов, расположенных в пределах небольшой территории. В настоящее время не существует четких ограничений на территориальный разброс абонентов локальной вычислительной сети. Обычно такая сеть привязана к конкретному объекту. К классу ЛВС относятся сети отдельных предприятий, фирм, банков, офисов, корпораций и т.д. Если такие ЛВС имеют абонентов, расположенных в разных помещениях, то они (сети) часто используют инфраструктуру глобальной сети Интернет, и их принято называть корпоративными сетями или сетями Интранет (Intranet).

Региональные сети связывают абонентов города, района, области или даже небольшой страны. Обычно расстояния между абонентами региональной ИВС составляют десятки-сотни километров.

Глобальные сети объединяют абонентов, удаленных друг от друга на значительное расстояние, часто находящихся в различных странах или на разных континентах. Взаимодействие между абонентами такой сети может осуществляться на базе проводных линий связи, систем радиосвязи и даже спутниковой связи.

Таблица 1.3. Сравнение сетей с коммутацией каналов и пакетов

Коммутация каналов	Коммутация пакетов
Необходимо предварительно устанавливать соединение	Отсутствует этап установления соединения (дейтаграммный способ)
Адрес требуется только на этапе установления соединения	Адрес и другая служебная информация передается с каждым пакетом
Сеть может отказать абоненту в установлении соединения	Сеть всегда готова принять данные от абонента
Гарантированная пропускная способность для взаимодействующих абонентов	Пропускная способность сети для абонентов неизвестна, задержки передачи носят случайный характер
Трафик реального времени передается без задержек	Ресурсы сети используются эффективно при передаче пульсирующего трафика
Высокая надежность передачи	Возможные потери данных из-за переполнения буферов
Нерациональное использование пропускной способности каналов, снижающее общую эффективность сети	Автоматическое динамическое распределение пропускной способности физических каналов в соответствии с фактической интенсивностью трафика абонентов

По скорости передачи информации сети подразделяются на низкоскоростные (до 10 Мбит/с), среднескоростные (до 100 Мбит/с) и высокоскоростные (> 100 Мбит/с). Наибольшими скоростями отличаются локальные сети. Следует заметить, что представленная выше классификация сетей по скоростям и, тем более граничные значения скоростей весьма условны.

По способу коммутации сети делятся на сети с коммутацией каналов и коммутацией пакетов. Сравнение этих сетей представлено в табл. 1.3 [8].

Сегодня предпочтение отдается технологии коммутации пакетов. Эта технология используется в сетях следующего поколения (Next Generation Network, NGN) — сетях, обеспечивающих предоставление неограниченного набора услуг с гибкими возможностями по их управлению, персонализации и созданию новых услуг. Сети NGN относятся к классу мультисервисных сетей [9].

Топология. Одной из основных характеристик сети, отражающей ее особенности, является ее топология. Сетевые узлы соединяются звеньями или линками (links), точное число линков относительно числа узлов определяется топологией сети. Топология определяет связность сети; чем больше линков на узел, тем лучше связность. Различают пять основных топологий: кольцо, звезда, mesh (ячеистая), дерево и шина (рис. 1.26) [2].

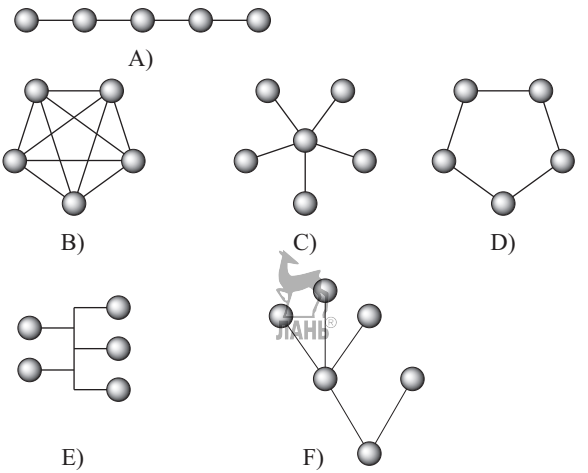


Рис. 1.26. Основные топологии

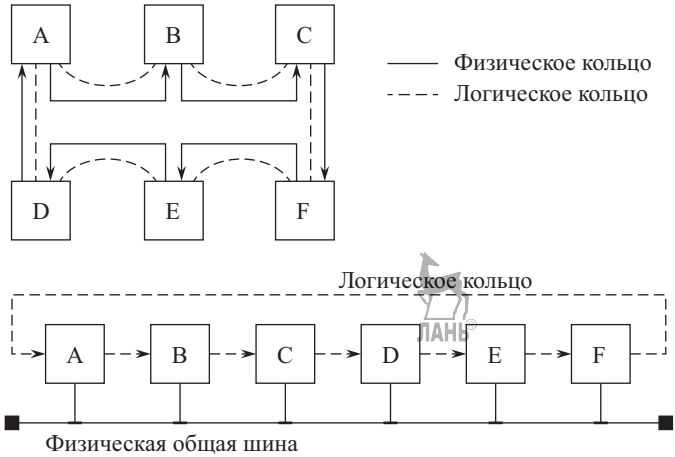


Рис. 1.27. Физическая и логическая топологии

Следует различать логическую и физическую топологию сети. Физическая инфраструктура определяется географическим расположением узлов и их соединениями. Логическая топология относительно соединений между узлами может существенно отличаться от физической и дает представление о путях, по которым передаются потоки информации между узлами (рис. 1.27).

Выбор топологии сети является важнейшей задачей, решаемой при ее построении, и определяется требованиями к экономичности и структурной надежности.

Кольцо является простейшей топологией, поддерживающей защиту от отказов, поскольку на кольце всегда создается два раздельных пути между узлами.

Ячеистая топология. В случае полносвязной ячеистой топологии, каждый узел соединен непосредственно со всеми другими узлами сети. Это обеспечивает хорошую отказоустойчивость. С другой стороны, такая топология требует число линков, пропорциональное квадрату числа узлов, что для больших сетей неосуществимо. Ячеистая сеть не обязательно должна быть полносвязной. В существующих ячеистых сетях не все узлы соединены друг с другом. С другой стороны, степень связности здесь более высокая, чем в кольцевых сетях, т.е. в среднем в таких сетях число путей между двумя узлами превышает два. Использование ячеистой топологии целесообразно на сетях с высокой концентрацией трафика.

В топологии *звезда* имеется центральный узел, который соединен со всеми другими узлами сети. Так как весь трафик должен проходить через центральный узел, то он должен иметь очень большую пропускную способность (или емкость — capacity). Более того выход из строя (отказ) центрального узла приводит к выходу из строя всей сети. Однако эта топология используется довольно часто (например, в локальных сетях) вследствие своей простоты и легкости расширения сети.

Шина является общей средой для всех узлов. Это делает эту топологию уязвимой к отказам на кабеле. Более того, необходимость обеспечения множественного доступа к среде требует использования специальных мер по недопущению конфликтов.

Дерево является идеальной топологией для распределенных (distributed) сетей. Она относительно дешевая, хорошо масштабируема. В оптических сетях распределение потоков может быть реализовано с использованием недорогих пассивных сплиттеров. Эта топология широко используется в оптических сетях доступа на базе технологии PON (Passive Optical Network).

Особенностью сегмента сети, имеющего древовидную топологию, является то, что связность n узлов на уровне физической топологии здесь достигается числом ребер $l = n - 1$, что обеспечивает высокую экономичность сети. На логическом уровне количество путей передачи информации между каждой парой узлов в таком сегменте всегда равно $P = 1$, что не всегда позволяет обеспечить требования к надежности сети. Повышение надежности в таких сетях достигается введением резервных связей.

Существует традиционное деление сетей на магистральные (core), городские (metro) и доступа (access). Каждый домен характеризуется собственной топологией (рис. 1.28).

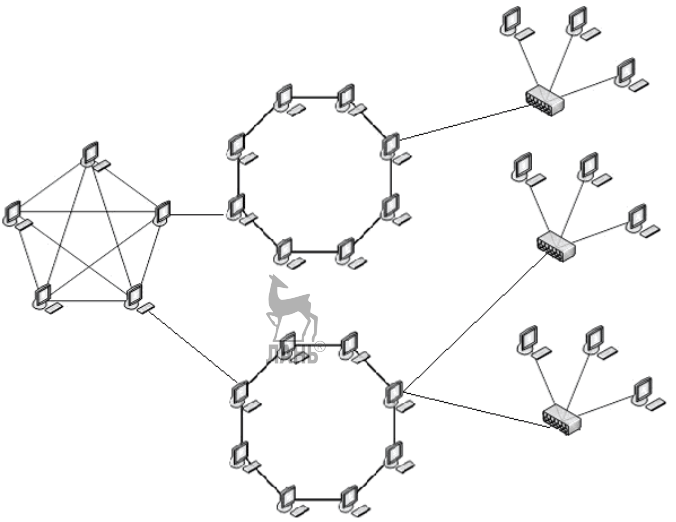


Рис. 1.28. Виды топологии, встречающиеся в доменах

1.7. Организация работ по стандартизации в области передачи данных

Стандарты и рекомендации. Существуют понятия «рекомендация» и «стандарт». *Стандарт* (от англ. standard — норма, образец, мерило) — это образец, эталон, принимаемый за исходный для сопоставления с ними других объектов. Нормативно-технический документ по стандартизации устанавливает комплекс норм, правил, требований к объекту стандартизации и утвержденный компетентным органом. *Рекомендация* — это совет, который, если ему следует большинство, принимается в качестве стандарта. Поскольку придерживаться стандартов и рекомендаций — дело добровольное, не существует международных законов, которые принуждали бы поставщиков оборудования следовать им. Однако игнорирование рекомендаций серьезно затруднит выход на рынок или даже сделает это невозможным. Недаром в описании той или иной телекоммуникационной продукции можно встретить упоминание о том, что она удовлетворяет тем или иным рекомендациям и стандартам. Рекомендации и стандарты разрабатываются как международными, так и национальными организациями. По определению, рекомендация — это нечто менее важное, чем стандарт, но на практике ее учитывают так же, как и стандарт. Продукт, быстро завоевывающий рынок, получает статус фактического стандарта. В этом случае говорят о стандарте «де факто». Когда продукт захватывает весь рынок, другие поставщики вынуждены приспосабливаться к фактическому стандарту.

В стандартизации заинтересованы не только потребители, но и пользователи: поскольку при этом появляется возможность использовать одновременно оборудование различных фирм, а выпуск одного и того же стандартного оборудования разными фирмами приводит к конкуренции, делая продукт менее дорогим. И пользователи, и поставщики получают выгоду от стандартизации. Поставщики могут конкурировать на равных условиях и при равных шансах на успех, поскольку пользователи совмещают оборудование различных поставщиков.

Работы по стандартизации сетей передачи данных ведутся большим количеством организаций. В зависимости от статуса организаций различают следующие виды стандартов:

- *стандарты отдельных фирм* (например, стек протоколов DECnet компании Digital Equipment или графический интерфейс OPEN LOOK для Unix-систем компании Sun);
- *стандарты специальных комитетов и объединений*, создаваемых несколькими фирмами, например стандарты технологии ATM, разрабатываемые специально созданным объединением ATM Forum, насчитывающим около 100 коллективных участников, или стандарты союза Fast Ethernet Alliance по разработке стандартов 100 Мбит Ethernet;
- *национальные стандарты*, например, стандарт FDDI — один из многочисленных стандартов, разработанных Американским национальным институтом стандартов (ANSI), или стандарты безопасности для операционных систем, разработанные Национальным центром компьютерной безопасности (NCSC) Министерства обороны США;
- *международные стандарты*, например, модель и стек коммуникационных протоколов Международной организации по стандартизации (ISO), многочисленные стандарты Международного союза электросвязи (ITU), в том числе стандарты на сети с коммутацией пакетов X.25, сети Frame Relay, ISDN, модемы и многие другие [8].

Некоторые стандарты, непрерывно развиваясь, могут переходить из одной категории в другую. В частности, фирменные стандарты на продукцию, получившую широкое распространение, обычно становятся международными стандартами, так как вынуждают производителей из разных стран следовать фирменным стандартам, чтобы обеспечить совместимость своих изделий с этими популярными продуктами. Например, из-за широкого распространения персонального компьютера компании IBM фирменный стандарт на архитектуру IBM PC стал международным стандартом.

Международные организации по стандартизации. Международный союз электросвязи (МСЭ) — самая старая организация объединенных наций, основанная в 1865 г., это самый важный международный орган по стандартизации в телекоммуникациях. Каждая страна, входящая в состав объединенных наций, имеет право стать членом МСЭ. Администрации, действующие операторы сети, научные, промышленные и международные организации могут принимать участие в работе МСЭ, который финансируется добровольными членскими взносами. В основные задачи МСЭ входят поддержка и развитие международного сотрудничества, активная поддержка развития технологий, техническое содействие развитию стран в сфере телекоммуникаций. Кроме того, МСЭ публикует руководства к ряду основных правил с целью помочь операторам сети в долгосрочном планировании. Эти правила распространяются на такие сферы как передача, сигнализация, маршрутизация, нумерация, синхронизация, тарификация и качество обслуживания.

С марта 1993 г. МСЭ разделен на три сектора: сектор телекоммуникаций МСЭ-Т (ITU-T), радиосвязи МСЭ-Р (ITU-R) и развития (ITU-D). Сектор МСЭ-Т отвечает за международную координацию всей телекоммуникационной области и с этой целью выпускает разнообразные стандарты, МСЭ-Т заменил МККТТ, который сейчас не функционирует. Практическую работу выполняют 15 главных исследовательских групп, состоящих из экспертов по телекоммуникациям стран-участниц (рис. 1.29).

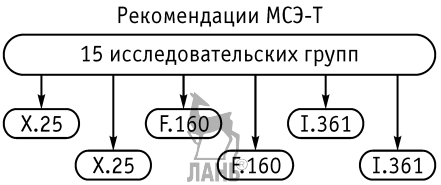


Рис. 1.29. Исследовательские группы МСЭ-Т

Стандарты компонуются в серию, обозначаемую буквой и числом; например, X.25. Часто ответственность за отдельную специализированную серию несет одна исследовательская группа. Примеры стандартов: X.25 — протокол для сетей с пакетным режимом передачи; G.803 — архитектура транспортной сети синхронных цифровых иерархий.

В состав МСЭ-Т входят следующие исследовательские группы:

- разделение служб;
- работа сети;
- принципы тарификации и учета;

- поддержка сети;
- защита от электромагнитных воздействий;
- выносное оборудование;
- сети передачи данных и открытые системы;
- терминалы для телематических служб;
- телевизионная и звуковая передача;
- языки для прикладных телекоммуникационных программ;
- коммутация и сигнализация;
- передача «из конца в конец», осуществляемая сетями и терминалами;
- основные аспекты сети;
- модемы и технические приемы передачи данных;
- телеграфные и телематические услуги;
- системы передачи.

Раньше рекомендации утверждались и публиковались в книжной форме каждые четыре года, позднее за этим последовал каталог «Голубая Книга», опубликованный в 1988 г. Поскольку для быстрого развития четыре года — это очень большой срок, МСЭ решил опубликовывать стандарты каждой исследовательской группы по мере их создания.

Сектор ITU-R отвечает за распределение радиочастот. Основная задача ITU-D — в том, чтобы с одной стороны брать на себя обязанности ITU, связанные с проектами развития в области телекоммуникаций; с другой — управлять финансированием объединенных проектов в развитых странах.

Кроме МСЭ к основным организациям по международной стандартизации относятся:

- ISO (International Organization of Standardization) — Международная организация по стандартизации;
- IEC (International Electrotechnical Committe) — Международный электротехнический комитет.

Международные организации иногда не успевают за развитием новых технологий и новых рыночных требований, поэтому на некоторых территориях организуются специальные заинтересованные группы, разрабатывающие отдельные рекомендации (рис. 1.30).

Эти группы предназначены для ускорения работ по стандартизации и облегчения внедрения продуктов и услуг. Одним из примеров является форум ATM — неофициальный орган, организованный для содействия внедрению оборудования, основанного на технологии асинхронного переноса информации. На практике форум ATM — организация, совместно с ITU-T развивающая стандарты ATM. Организации такого типа часто создаются тогда, когда на рынке телекому-

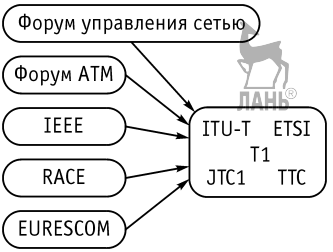


Рис. 1.30. Взаимодействие неофициальных и официальных организаций

никаций появляются новые действующие лица, не связанные с ITU. Другим примером организаций, возникших для содействия развитию в определенной сфере, являются форумы управления сетью и коммуникациями. Также как и ITU, они финансируются их членами.

Особую роль в выработке международных открытых стандартов играют стандарты Internet. Ввиду постоянно растущей популярности Internet, эти стандарты становятся международными стандартами «де-факто», и многие из них приобретают впоследствии статус официальных международных стандартов. Существует несколько организационных подразделений, отвечающих за развитие Internet.

Основным из них является «Общество Интернета» (Internet Society, ISOC) — профессиональное сообщество, которое занимается общими вопросами эволюции и роста Internet как глобальной коммуникационной инфраструктуры. Под управлением ISOC работает Internet Architecture Board (IAB) — организация, в ведении которой находится технический контроль и координация работ для Internet. IAB координирует направление исследований и новых разработок для стека TCP/IP и является конечной инстанцией при определении новых стандартов Internet. Информационные документы Интернета, содержащие технические спецификации и стандарты, публикуются в виде RFC (Request for Comments). В переводе на русский язык RFC «тема для обсуждения». Документы RFC издаются на английском языке. Некоторые из них переведены на русский язык (см. www.protocols.ru).

Региональные организации. В некоторых частях мира, в основном в Европе, США и Японии, сформированы региональные организации по стандартизации. Подобные организации могут иметь очень большое значение. Возьмем для примера ANSI (American National Standards Institute) или ETSI (European Telecommunications Standards Institute). Официальная Европейская организация по стандартизации телекоммуникаций ETSI создана в 1988 г., выпускает рекомендации для общего европейского рынка, отличается от своего предшественника CEPT (Европейская конференция почтовых и телекоммуникацион-

ных ведомств) тем, что имеет более широкую членскую базу. Раньше работа по стандартизации в Европе осуществлялась CEPT, членами которого были телекоммуникационные администрации с монопольным статусом. Членами ETSI являются администрации связи, операторы сети, поставщики и производители услуг, пользователи — все эти категории сейчас непосредственно влияют на работу по стандартизации. Сегодня все стороны заинтересованные в стандартизации, участвуют в ней на равных условиях. Организация ETSI состоит из нескольких технических комитетов, каждый из которых разделен на несколько подкомитетов. Организуются и специальные рабочие группы для создания отдельных рекомендаций. Эти рабочие группы состоят из специалистов — членов различных организаций, приглашаемых на определенный срок. Работа осуществляется как проект в течение ограниченного периода времени. Важным моментом работы ETSI является создание международного адаптированного стандарта, имеющего отношение к продуктам телекоммуникаций для использования на внутреннем европейском рынке.

Кроме европейских, к организациям по локальной стандартизации относятся: TI — Комитет телекоммуникаций (США); TIA — Ассоциация промышленных телекоммуникаций (США); IEEE — Институт инженеров по электронике и электротехнике (США); TTC — Комитет телекоммуникационных технологий (Япония); RSR — Центр исследования и развития радиосистем (Япония).

Национальные организации. Национальный стандарт часто базируется на региональных рекомендациях. Во многих случаях они переносятся в стандарт непосредственно, в других случаях приспосабливаются к среде и условиям каждой страны. Выпускаемых национальными организациями инструкций становится все больше. Они имеют узкую специализацию, при этом принимаются во внимание детали и предложения международных рекомендаций. Российские стандарты телекоммуникаций разрабатываются в соответствии с рекомендациями ITU-T и TTSI.

1.8. Эталонная модель взаимодействия открытых систем

При передаче речи через телекоммуникационную сеть придерживаются определенных правил. Вы представляете себя, называя свое имя (как при рукопожатии), повторяете слова и предложения, когда Ваш разговор прерван, возобновляете разговор после повреждения и заканчиваете заключительной фразой. Эти правила могут называться протоколами в значении слова, используемого в мире телекоммуникаций. Сетевой оператор предпочитает иметь одну эффективную по стоимости сеть вместо разных взаимодействующих фиксированных и мобильных сетей и служб, обеспечивающих передачу речи, видео и

данных.

Современные сети можно рассматривать как синтез двух исходно независимых сетей — телекоммуникационных и компьютерных. Логика развития систем связи требовала применения цифровых систем передачи (ЦСП) и вычислительных средств для решения задач маршрутизации, управления установлением соединений, сигнализации, а логика развития вычислительной техники побуждала к большему использованию средств связи между периферийными устройствами и отдельными компьютерами. Требования, предъявляемые к системам телекоммуникаций (СТ), сводятся к обеспечению высококачественной передачи, распределению, обработке и хранению разнородной информации, возможности управления со стороны пользователя, оперативного получения от сети ответных реакций, объединения и разделения ресурсов. Эти требования могут быть обеспечены лишь на основе полной сопряженности, осуществляемой в рамках распределенных процессов управления и обработки информации.

В 70-х гг. каждая из двух самых больших компьютерных корпораций развивались по своему собственному стандарту. Корпорация Digital Equipment называла свой стандарт DNA (цифровая сетевая архитектура), в то время как стандарт корпорации IBM назывался SNA (системная сетевая архитектура). Из-за отсутствия общего стандарта архитектуры этих корпораций были несовместимы, выпускаемое ими оборудование не могло взаимодействовать друг с другом. Для решения этой проблемы были необходимы стандарты, позволяющие соединяться системам производства различных корпораций. Качественные изменения в технике связи потребовали проведения интенсивной работы МСЭ и ISO по созданию унифицированной модели взаимодействия распределенных процессорных систем и выработке международных стандартов. Был создан подкомитет ISO для разработки международных стандартов взаимодействия открытых систем (ВОС). Термин «открытая система» подразумевает систему, взаимодействующую с любой другой системой, удовлетворяющей тем же стандартам. Открытая система открыта для развития как количественно, так и качественно, она гибкая, так как допускает эволюцию с учетом новых теоретических и технических возможностей. Объединение открытых систем также является открытой системой. Работа по стандартизации открытых систем началась в 1977 г. В 1983 г. была предложена эталонная модель ВОС — наиболее общее описание структуры построения стандартов. Модель ВОС, определяющая принципы взаимосвязи между отдельными стандартами, является основой для параллельной разработки множества стандартов и обеспечивает постепенность перехода от существующих реализаций к новым стандартам. Модель

ВОС была принята в 1984 г. и опубликована в «Красной книге» как рекомендация X.200.

При создании стандарта установления связи между двумя ПК для передачи данных должно быть определено несколько «соглашений»:

- какая служба и какой язык будут использоваться;
- как будет кодироваться информация;
- как будет отображен диалог, как он будет начинаться и заканчиваться;
- что должен сделать пользователь, чтобы определить нарушение передачи;
- какие методы будут использоваться для «присоединения» адреса;
- как будет осуществляться контроль сети при нарушении передачи;
- как терминалы будут соединяться в сети.

Разбиение на уровни является механизмом, используемым для разбиения трудоемкой задачи на подзадачи, и должно производиться исходя из следующих соображений:

- уровней не должно быть слишком много, чтобы разработка сети и ее реализация не были чрезмерно сложными. В то же время, если уровней будет слишком мало, функции, выполняемые каждым уровнем, будут чрезмерно сложными;
- новый уровень должен создаваться по мере необходимости создания отдельного уровня абстракции;
- каждый уровень должен выполнять строго определенную функцию;
- выбор функций для каждого уровня должен осуществляться с учетом создания стандартизованных международных протоколов;
- количество информации, передаваемое через интерфейсы между уровнями, должно быть минимальным.

В общем случае, в соответствии с моделью OSI, сеть должна иметь семь функциональных уровней. В семиуровневой модели ВОС все процессы, реализуемые открытой системой, разбиты на взаимоподчиненные уровни. Уровень с меньшим номером предоставляет услуги смежному с ним верхнему уровню и пользуется для этого услугами нижнего смежного уровня. Внутренние функции на каждом уровне могут реализоваться различными средствами и по различным алгоритмам, однако взаимодействие между уровнями и компонентами одного уровня отдельных систем должно быть стандартным. На верхних уровнях располагаются прикладные процессы, а нижние отражают функции передачи информации различного вида в сети свя-

зи. *Прикладными процессами* называются программные компоненты, выполняющие обслуживание пользователей сети: передачу файлов, информационно-справочные услуги, передачу данных, организацию распределенного банка данных, диалоговую работу с пакетами прикладных программ, электронную почту, проведение телеконференций и т.п.

Основные понятия модели ВОС — протокол и интерфейс. Эталонная модель ВОС не определяет протоколы и интерфейсы взаимодействия, структуру и характеристики физических средств соединения. Она определяет лишь требования к ним и дает четкое описание характеристик области взаимодействия открытых систем, в рамках которых могут быть разработаны протоколы, интерфейсы и физические средства. В одной и той же эталонной модели для различных применений может быть описано множество наборов услуг, каждый из которых удовлетворяет требованиям ВОС. Последней ступенью детализации является разработка набора протоколов в рамках определенных услуг. Для каждого набора услуг могут быть разработаны различные протоколы. Таким образом, стандарт ВОС должен определять не только эталонную модель, но и конкретный набор услуг, удовлетворяющих этой модели, а также набор протоколов, обеспечивающих предоставление этих услуг. Система является открытой лишь тогда, когда она соответствует эталонной модели ВОС, стандартному набору услуг и стандартным протоколам.

Разбиение на уровни выполнено в соответствии со следующими принципами (рис. 1.31).

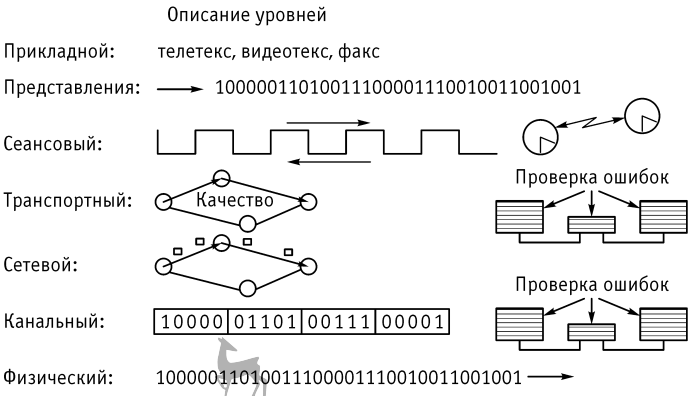


Рис. 1.31. Описание уровней

Верхний (седьмой) уровень модели ВОС является основным, ради которого существуют все остальные уровни. Он называется *прикладным*, поскольку с ним взаимодействуют прикладные процессы

данной системы, решающие некоторую задачу совместно с другими прикладными процессами, размещенными в других открытых системах. Протоколы прикладного уровня обеспечивают взаимодействие прикладных процессов и непосредственно связаны с пользовательскими программами. Этот уровень эталонной модели ВОС определяет семантику, т.е. смысловое содержание информации, которой обмениваются открытые системы в процессе совместного решения некоторой заранее известной задачи. Естественно, что обе взаимодействующие системы должны одинаково интерпретировать получаемую информацию, т.е. обладать представлениями об области совместной работы. Функции, реализуемые протоколами этого уровня, обеспечивают описание форм и методов взаимодействия прикладных процессов; идентификацию пользователей; посылку запросов на соединение с другими прикладными процессами; подачу заявок нижеследующему уровню на необходимые методы описания информации и т.д. Для того, чтобы прикладные процессы могли взаимодействовать между собой, необходимо соблюдение протокола о смысловом содержании всех затрагиваемых аспектов работы.

Следующий, шестой уровень, называется уровнем *представления*. Он определяет синтаксис передаваемой информации, т.е. набор знаков и способы их представления, понятные для всех взаимодействующих открытых систем. По протоколам уровня представления взаимодействующие системы договариваются о форме, в которой будет передаваться информация. Задача этого уровня — преобразование информации, подлежащей передаче между прикладными процессами, т.е. язык и формат представления информации: данных, графического материала или речи.

Пятый, *сеансовый* уровень, обеспечивает взаимодействие между прикладными процессами независимо от метода и техники передачи информации. Протоколы этого уровня вызывают необходимые пользовательские программы, выделяют ресурсы, необходимые для их выполнения, и обеспечивают связь с пользовательскими программами. Сеансовый уровень предоставляет в распоряжение пользователей средства для организации диалога между процессами двух верхних уровней. Пример выполняемых функций: открытие и закрытие сеанса связи; синхронизация сеансового соединения; аналоговое управление сеансом, обеспечивающее передачу блоков данных и подтверждение правильности приема. При взаимодействии прикладных процессов, реализованных в одной системе, сеансовый уровень является самым нижним.

Четвертый, *транспортный* уровень, обеспечивает логическое соединение между двумя оконечными устройствами от одного пользо-

вателя к другому согласно адресам источника и получателя сообщений и пересылку сообщений между взаимодействующими системами с использованием нижних уровней. Этот уровень принимает от верхнего уровня некоторый блок данных и обеспечивает его транспортировку через сеть связи к удаленной системе с требуемым качеством обслуживания. Уровни, лежащие выше транспортного, не учитывают специфику сети, через которую передаются данные, они знают лишь удаленные системы, с которыми взаимодействуют. Транспортный уровень должен иметь информацию о том, как работает сеть, размеры блоков данных, которые она может принимать и т.п. Функции четвертого уровня включают в себя также процедуры контроля и коррекции ошибок. Все вышеперечисленные уровни называются верхними и реализуются только в системах ВОС.

Три следующих нижних уровня определяют функционирование узлов сети и они должны реализовываться в системах, взаимодействующих через канал связи с узлом сети или другой открытой системой.

Третий, *сетевой* уровень, выполняет маршрутизацию блоков данных через сеть. Он включает в себя функции установления физического соединения последовательно через все звенья сети и содержит сигнальный протокол, определяющий маршрут передачи информации от одного объекта к другому. Протоколы этого уровня реализуют функции выбора маршрутизации и типа коммутации.

Второй, *канальный* уровень, выполняет функции установления, поддержания и разъединения соединений каналов связи. Эти соединения называют информационными каналами. Для обеспечения информационных каналов может выполняться разделение информации на отдельные сегменты, называемые блоками, кадрами или пакетами. Каждый блок информации может содержать поля фиксированного размера (адреса, управления и проверки) и поле переменной длины (информационное). Протоколы канального уровня реализуют оптимальную длину блока. В целом на канальный уровень возлагаются следующие функции:

- *инициализации* — обмена служебными сообщениями, подтверждающими готовность к передаче данных, между взаимодействующими узлами сети;
- *идентификации* — обмена служебной информацией, подтверждающей правильность соединения между пунктами;
- *синхронизация* по кодовым комбинациям;
- *сегментация* — формирование блоков для их передачи по каналу;
- *обеспечение прозрачности* — предоставления вышераспо-

ложенному уровню возможности передачи произвольных последовательностей битов или знаков;
- *управления потоком* — обеспечение согласования скоростей передачи и приема;
- *контроля ошибок* в канале связи и восстановления информации, искаженной в процессе передачи по сети;
- *обнаружения нарушений* нормальной передачи информации и реализации процедур выхода из сбойных ситуаций;
- *ликвидации* логического соединения, образованного при инициализации канала;
- *управления каналом* — обеспечения возможности контроля функционирования канала, выявления отказов, восстановления, сбора информации о работе канала. Услуги канального уровня различны для разных информационных каналов.

Первый, *физический* уровень, обеспечивает непосредственную взаимосвязь со средой передачи, реализуя механические, электрические, функциональные и процедурные стандарты взаимодействия с физическими средствами ПД. Примеры среды передачи: коаксиальный кабель, двухпроводная витая пара, световод, шина, состоящая из группы проводов, для параллельной передачи байтов информации и др. Среда передачи может быть составной и включать сегменты различного типа, например, проводную и световодную линии. При этом в функции данного уровня не входит создание самой физической среды передачи, он определяет только основные характеристики потока информации через эту среду, например, скорость передачи, вид синхронизации. Целью физического уровня является установление, поддержание и отключение физических соединений (физических каналов), соединяющих между собой узлы сети. Физические уровни узлов сети, соединенных каналами с различной средой передачи, должны выполнять одинаковые протоколы взаимодействия. Например, оконечный электронно-оптический преобразователь должен взаимодействовать с системой своего узла так же, как и модем, расположенный на другом узле сети. Правила этого взаимодействия и определяются протоколами физического уровня, реализованными на интерфейсе физического уровня со средой передачи. Сам интерфейс физического уровня представляет собой группу проводов для передачи в каждом направлении данных и управляющих сигналов.

В соответствии с архитектурой открытых систем, физический уровень должен предоставлять канальному уровню следующие услуги:
- реализовать физическое соединение между двумя или более компонентами канального уровня;

- осуществлять передачу по установленному соединению единиц данных, например, битов или байтов при передаче информации;
- предоставлять канальному уровню доступ к соединению, выполненному на физическом уровне;
- обеспечивать идентификацию путей передачи информации между компонентами физического уровня;
- обеспечивать сохранение на выходе той же последовательности данных, которая поступала на вход физического соединения;
- выдавать сообщения об отказах или неисправных состояниях физического уровня;
- обеспечивать требуемые параметры качества обслуживания.

Таким образом, три нижних уровня обеспечивают непосредственно транспортировку информации от источника к потребителю. При рассмотрении модели ВОС следует иметь в виду, что число используемых уровней может быть различным для фазы установления соединений и фазы непосредственного обмена информацией между пользователями.

В отличие от эталонной модели ВОС, протокольные модели конкретных сетей допускают введение дополнительных подуровней, а также могут включать не все уровни. Однако их построение основывается на тех же принципах. Модель ВОС главным образом предназначена для координации существующих и будущих стандартов. Стандарты, применяющиеся на отдельных уровнях, определяют другие рекомендации. Некоторые примеры таких стандартов: соединение терминалов данных с модемами (V.24), канальный протокол (LAPB), адресация (Q.931). Разделение служб на службы передачи и телеслужбы как раз отражает стандартизацию функций и протоколов в модели ВОС. Так как службы передачи ориентированы на транспортировку сообщений, они реализуются на нижних уровнях модели ВОС. Телеслужбы охватывают все без исключения функции передачи и протоколы связи для всех уровней модели ВОС.

Соединение двух компьютерных систем, в которых реализована семиуровневая модель взаимодействия открытых систем, представлено на рис. 1.32. Передающая сторона генерирует данные, используемые на приемной стороне. На каждом уровне модели ВОС последовательно добавляются данные, используемые соответствующим уровнем на приемной стороне. Эти данные обычно располагаются в форме заголовка в начале пакета данных, доставляемого от вышестоящего уровня. На сетевом уровне данные часто добавляются в конце пакета в форме трейлера (хвоста). Дополнительные биты вставляются даже

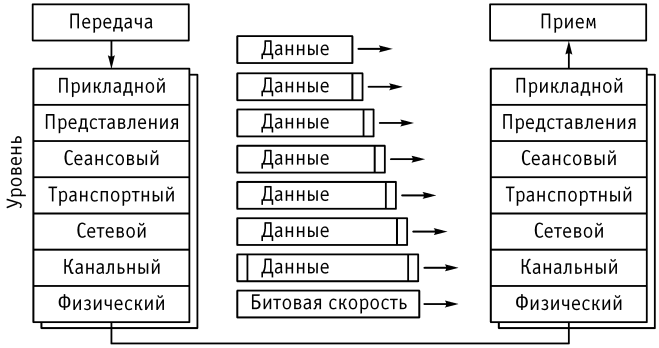


Рис. 1.32. Передача данных в модели ВОС

на физическом уровне, работающем в цифровом режиме.

Каждый уровень ВОС имеет свой собственный протокол формата данных. Хорошо продуманная структура создается модульно, это означает, что протокол одного уровня может быть заменен без повреждения других уровней. Первый–третий уровни являются функциональными эквивалентами сетевых узлов. Узлы должны быть предоставлены как можно меньшим числом уровней, функции некоторых уровней могут быть совмещены.

1.9. Контрольные вопросы

1. Приведите определение терминов «информация», «сообщение», «сигнал».
2. От чего зависит количество информации, которое содержится в сообщении?
3. Различие понятий «скорость модуляции» и «скорость передачи информации».
4. Как обеспечить передачу информации со скоростью $R > B$?
5. Пояснить, почему с ростом скорости передачи информации помехоустойчивость падает?
6. Перечислите общие характеристики физической среды, не зависящие от их физической природы.
7. Категории витой пары.
8. Чем отличаются кабели вида UTP от кабелей STP?
9. Перечислите достоинства и недостатки ВОЛС.
10. Сущность и назначение методов линейного кодирования.
11. Для чего используются логические линейные коды?
12. Виды модуляции.
13. Особенности OFDM.
14. Достоинства и недостатки OFDM.

15. Для чего используются методы доступа к среде CSMA/CD и CSMA/CA?
16. Пояснить принцип частотного разделения каналов.
17. Пояснить принцип временно́го разделения каналов.
18. Достоинства спектрального волнового уплотнения каналов.
19. Как осуществляется доступ к среде при использовании OFDMA?
20. За счет чего обеспечивается разделение каналов при использовании CDMA?
21. Пояснить сущность терминов «услуга» и «служба».
22. Классификация сетей электросвязи по территориальной распространенности.
23. Виды топологии сетей, их достоинства и недостатки.
24. Чем отличаются понятия «рекомендация» и «стандарт»?
25. Из каких соображений должно выбираться число уровней разбиения трудоемкой задачи на подзадачи?
26. Сколько уровней содержит модель ВОС?
27. Что такое протокол и интерфейс?
28. Дайте краткую характеристику уровней модели ВОС.

1.10. Список литературы

1. *Крук Б.И., Попантонопуло В.Н., Шувалов В.П.* Телекоммуникационные системы и сети. В 3-х томах. Том 1. Современные технологии / под ред. проф. В.П. Шувалова. — Изд. 4-е, испр. и доп. — М.: Горячая линия–Телеком, 2012. — 620 с.
2. *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. — 4-е изд. — СПб.: Питер. — 944 с.
3. *Катунин Г.П., Мамчев Г.В., Попантонопуло В.Н., Шувалов В.П.* Учебное пособие в 3-х томах. Том 2. Радиосвязь, радиовещание, телевидение / под ред. проф. В.П. Шувалова. — 3-е изд. стереотип. — М.: Горячая линия–Телеком, 2013. — 672 с.
4. Англо-русский словарь по вычислительной технике и информационным технологиям / Составитель С.Б. Орлов. — 4-е изд., перераб. и доп. — М.: ИП Радио-Софт, 2005. — 640 с.
5. Большой Российский энциклопедический словарь. — М.: БРЭ, 2003. — С.1437.
6. *Gopalakrishnan T.R., Nair Abhijith N., Sooda K.* Transformation of Networks through Cognitive Approaches // Journal of Research & Industry. — December 2008. — Vol. 1, Is. 1.
7. *Комышанский В.И., Соколов Н.А.* Когнитивные системы и телекоммуникационные сети // Вестник связи. — 2011. — №10. — С.4–8.

8. *Олифер В.Г., Олифер Н.А.* Основы компьютерных сетей. — СПб.: Питер, 2009. — 352 с.
9. *Величко В.В., Субботин Е.А., Шувалов В.П., Ярославцев А.Ф.* Телекоммуникационные системы и сети. В 3-х томах. Том 3. Мультисервисные сети / под ред. проф. В.П. Шувалова. — М.: Горячая линия–Телеком, 2005. — 592 с.
10. *Захарченко М.В. и др.* Сети и системы телекоммуникаций. В 4-х томах. Том 1 / под ред. проф. Н.В. Захарченко. — Киев: Техника, 2000. — 304 с.

Глава 2

Обеспечение показателей качества обслуживания

2.1. Качество обслуживания. Общие положения

Согласно [1], качество обслуживания (QoS) определяется как «совместный эффект производительности службы, который определяет степень удовлетворенности пользователя службы». Это определение QoS, несмотря на его абстрактность, довольно широко используется. Итак, необходимость рассмотрения QoS как показателя удовлетворения пользователей, является определяющим.

Методы обеспечения качества обслуживания (QoS) занимают сегодня важное место в арсенале технологий сетей с коммутацией пакетов, так как они обеспечивают устойчивую работу современных мультимедийных приложений, таких как высокоскоростная передача данных, IP-телефония, видео- и радиовещание, интерактивное дистанционное обучение и т.п. Методы обеспечения QoS направлены на улучшение характеристик производительности и надежности сети; эти методы позволяют уменьшить задержки, вариации задержек, а также потери пакетов в периоды перегрузки сети, создавая тем самым необходимые условия для удовлетворительного обслуживания сетью трафика приложений.

Методы обеспечения качества обслуживания направлены на компенсацию негативных последствий временных перегрузок, возникающих в сетях с коммутацией пакетов. В этих методах используются различные алгоритмы управления очередями, резервированием и обратной связью, позволяющие снизить негативное влияние очередей до приемлемого для пользователей уровня.

Характеристики качества обслуживания. Учет характеристик QoS особенно важен в том случае, когда сеть обеспечивает передачу одновременно разного типа трафика, например, трафика веб-приложений и голосового трафика. Это связано с тем, что различные типы трафика предъявляют разные требования к характеристикам QoS. Добиться одновременного соблюдения всех характеристик QoS для всех видов трафика очень сложно. Поэтому обычно используют следующий подход: классифицируют все виды трафика, существующие в сети, относя каждый из них к одному из распространенных типовых видов трафика, а затем добиваются одновременного выполнения определенного подмножества из набора требований для этих типов трафика.

Таблица 2.1. Нормы для характеристик сетей IP с распределением по классам качества обслуживания

Сетевые характеристики	Классы QoS					
	0	1	2	3	4	5
Задержка доставки пакета IP, IPTD	100 мс	400 мс	100 мс	400 мс	1 с	Н
Вариация задержки пакета IP, IPDV	50 мс	50 мс	Н	Н	Н	Н
Коэффициент потери пакетов IP, IPLR	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	Н
Коэффициент ошибок пакетов IP, IPER	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	Н

Примечание: Н — не нормировано

Качество обслуживания является предметом активных исследований и стандартизации на протяжении всей истории развития телекоммуникаций. Существенный вклад в развитие различных аспектов концепции QoS внес Международный союз электросвязи (МСЭ), включая, в том числе, разработку норм и требований к показателям качества обслуживания, стандартизации сетевых механизмов, обеспечивающих необходимые показатели QoS, а также формулировку основополагающих определений.

Так, например, в Рекомендации Y.1540 рассматриваются следующие сетевые характеристики как наиболее важные по степени их влияния на сквозное качество обслуживания (от источника до получателя), оцениваемое пользователем:

- производительность сети;
- надежность сети/сетевых элементов;
- задержка;
- вариация задержки (джиттер);
- потери пакетов.

Рекомендация МСЭ Y.1541 определяет нормы для параметров, определенных в Рекомендации Y.1540, между двумя граничными сетевыми интерфейсам — точками подключения оконечных терминальных устройств. Кроме того, в этой рекомендации специфицированы шесть классов качества обслуживания в зависимости от приложений и сетевых механизмов, применяемых для обеспечения гарантированного качества обслуживания. В табл. 2.1 представлены нормы на определенные сетевые характеристики для каждого из шести классов QoS.

Рекомендация Y.1541 устанавливает соответствие между классами качества обслуживания и приложениями:

- Класс 0 — Приложения реального времени, чувствительные к

джиттеру, характеризуемые высоким уровнем интерактивности (VoIP, видеоконференции);
- Класс 1 — Приложения реального времени, чувствительные к джиттеру, интерактивные (VoIP, видеоконференции);
- Класс 2 — Транзакции данных, характеризуемые высоким уровнем интерактивности (например, сигнализация);
- Класс 3 — Транзакции данных, интерактивные;
- Класс 4 — Приложения, допускающие низкий уровень потерь (короткие транзакции, массивы данных, потоковое видео);
- Класс 5 — Традиционные применения сетей IP.

Обзор методов обеспечения качества обслуживания. В методах, направленных на обеспечение качества обслуживания используются различные механизмы, способствующие снижению негативных последствий пребывания пакетов в очередях с сохранением в то же время положительной роли очередей. Набор механизмов достаточно широк. Большинство из них учитывает и использует в своей работе факт существования в сети трафика различного типа в том отношении, что каждый тип трафика представляет различные требования к характеристикам производительности и надежности сети. Например, трафик просмотра веб-страниц мало чувствителен к задержкам пакетов и не требует гарантированной пропускной способности сети, зато чувствителен к потерям пакетов; в то же время голосовой трафик очень чувствителен к задержкам пакетов, требует гарантированной пропускной способности сети, но может «терпеть» потерю небольшого процента пакетов без значительного ущерба для качества (впрочем, последнее свойство во многом зависит от используемого метода кодирования голосового сигнала).

Добиться одновременного соблюдения всех характеристик QoS для всех видов трафика весьма сложно. Одним из наиболее значимых факторов, влияющих на характеристики качества обслуживания, является уровень загрузки сети трафиком, то есть уровень использования пропускной способности каналов связи.

Если этот уровень постоянно достаточно низок, то трафик всех приложений обслуживается с высоким качеством бóльшую часть времени (хотя кратковременные перегрузки сети, приводящие к задержкам и потерям пакетов, все равно возможны, но они случаются очень редко). Такое состояние сети называется «недогруженным» или же используется термин «сеть с избыточной пропускной способностью» (англоязычный термин overprovisioning) [2]. Постоянно поддерживать все части сети в недогруженном состоянии достаточно дорого и сложно, но для наиболее ответственной части сети, такой как магистраль, этот подход применяется и связан он с постоянным слежением за

уровнем загрузки каналов магистрали и периодическим увеличением их пропускной способности по мере приближения загрузки к критическому уровню.

Методы обеспечения QoS основаны на другом подходе, а именно тонком перераспределении имеющейся пропускной способности между трафиком различного типа в соответствии с требованиями приложений. Очевидно, что это усложняет сетевые устройства, так как появляется необходимость знать требования всех классов трафика, уметь их классифицировать и распределять пропускную способность сети между ними. Последнее свойство обычно достигается за счет использования нескольких очередей пакетов для каждого выходного интерфейса коммуникационного устройства вместо одной очереди; при этом в очередях применяют различные алгоритмы обслуживания пакетов, чем и достигается дифференцированное обслуживание трафика различных классов. Поэтому методы QoS часто ассоциируются с техникой управления очередями.

Помимо собственно техники организации очередей, к методам QoS относят методы контроля параметров потока трафика, так как для гарантированно качественного обслуживания нужно быть уверенными, что обслуживаемые потоки соответствуют определенному профилю. Эта группа методов QoS получила название методов *кондиционирования* трафика.

Особое место занимают методы обратной связи, которые предназначены для уведомления источника трафика о перегрузке сети. Эти методы рассчитаны на то, что при получении уведомления источник снизит скорость выдачи пакетов в сеть и тем самым ликвидирует причину перегрузки.

Механизмы QoS можно применять по-разному. В том случае, когда они применяются к отдельным узлам без учета реальных маршрутов следования потоков трафика через сеть, условия обслуживания трафика этими узлами улучшаются, но гарантий того, что поток будет обслужен с заданным уровнем качества, такой подход не дает. Гарантии можно обеспечить, если применять методы QoS системно, резервируя ресурсы сети для потока на всем протяжении его маршрута, другими словами, «из конца в конец».

К методам QoS тесно примыкают методы *инжиниринга* трафика. Они обеспечивают управление маршрутами передачи данных таким образом, чтобы создать сбалансированную загрузку всех ресурсов сети и исключить за счет этого перегрузку коммуникационных устройств и образование длинных очередей. В отличие от методов QoS, в методах инжиниринга трафика не прибегают к организации очередей с различными алгоритмами обслуживания на сетевых

устройствах. В то же время в методах QoS в их традиционном понимании не используют такой мощный рычаг воздействия на рациональное распределение пропускной способности, как изменение маршрутов трафика в зависимости от фактической загрузки каналов связи, что позволяет легко отделить методы QoS от методов инжиниринга трафика.

В следующей группе методов борьба с перегрузками ведется путем снижения постоянной нагрузки на сеть. То есть в этих методах проблема рассматривается с другой стороны: если пропускной способности сети недостаточно для качественной передачи трафика приложений, то нельзя ли уменьшить объем самого трафика? Наиболее очевидным способом снижения объема трафика является его компрессия. Существуют и другие способы, приводящие к тому же результату, например, размещение источника данных ближе к его потребителю (кэширование данных).

Механизмы поддержки QoS в IP-сетях, IntServ. В середине 90-х годов начались работы по созданию стандартов QoS для IP-сетей, на основе которых можно было бы создать систему поддержки параметров QoS в масштабах составной сети и даже Интернета.

Процесс превращения сети Интернет в середине 90-х гг. из академической в коммерческую инфраструктуру, рост числа узлов и количества пользователей, применение сети Интернет для разнообразных приложений с различными требованиями к качеству обслуживания — все эти факторы определили быстрое развитие механизмов поддержки QoS. В ответ на новые условия, возникшие в сетях IP, Комитет IETF предложил большой набор моделей и механизмов для обеспечения качества обслуживания в сетях Интернет, которые разделяются на две категории в соответствии с названиями рабочих групп Комитета IETF, разрабатывающих эти модели и механизмы — интегрированных и дифференцированных услуг [3, 4].

Рабочая группа Integrated Services Working Group разрабатывала модель предоставления интегрированных услуг (или IntServ), основанную на принципе интегрированного резервирования ресурсов. Модель IntServ была разработана для поддержки приложений реального времени, чувствительных к задержкам. Механизмы, реализующие модель интегрированных услуг, должны обеспечивать взаимодействие всех сетевых устройств для поддержки любого уровня QoS вдоль пути передачи определенного потока пакетов.

Интегрированное обслуживание основано на резервировании ресурсов маршрутизаторов вдоль пути следования потока данных от одного конечного узла (точнее, приложения) до другого. Приложение должно использовать соответствующий прикладной программный ин-

терфейс API (Application Program Interface), чтобы передать запрос о резервировании ресурсов для определенного потока. Подобное резервирование является однонаправленным, так что если гарантированное качество обслуживания должно быть обеспечено для двустороннего обмена, потребуются две операции резервирования.

Механизм поддержки QoS в IP-сетях. DiffServ. Модель дифференцированных услуг (Differentiated Services, DiffServ) является логическим продолжением работ IETF над архитектурой IntSerf. Недостатки, заложенные в самом принципе модели IntServ (жесткие гарантии качества обслуживания, низкий уровень масштабирования) привели к необходимости создания более гибких механизмов обеспечения QoS. Общая характеристика принципов предоставления дифференцированных услуг [RFC-2475] [4] была опубликована в декабре 1998 г., а более детальные спецификации появились в середине 1999 г. Методы DiffServ составляют группу механизмов, которые в отличие от методов IntServ обеспечивают относительное или «мягкое» качество обслуживания.

Основная идея механизмов DiffServ состоит в предоставлении дифференцированных услуг для набора классов трафика (а не отдельных потоков), отличающихся требованиями к показателям качества обслуживания. Как и в случае механизмов IntServ, для реализации дифференцированных услуг широко применяются механизмы, входящие в состав рассмотренной выше архитектуры поддержки QoS в сетях IP.

Напомним, что классом трафика называется совокупность поступающих на обработку пакетов, обладающих общими признаками, например, все пакеты голосовых приложений или все пакеты с различными максимально допустимыми значениями длины поля данных кадров (Maximum Transmission Unit, MTU).

В отличие от потока в классах трафика, пакеты не различаются в зависимости от их маршрутов; это отличие иллюстрирует рис. 2.1. Так, маршрутизатор R1 относит все потоки, требующие приоритетного обслуживания и поступающие на его интерфейс i1, к одному классу, независимо от их дальнейшего маршрута. Маршрутизатор R2 оперирует уже другим составом приоритетного класса, так как в него вошли не все потоки интерфейса i1 маршрутизатора R1.

Обычно в сети DiffServ поддерживается дифференцированное обслуживание небольшого количество классов трафика, например, двух (чувствительного к задержкам и эластичного) или трех (к первым двум прибавляется класс, требующий гарантированной доставки пакетов с определенным минимумом скорости передачи трафика). Небольшое количество классов определяет масштабируемость этой

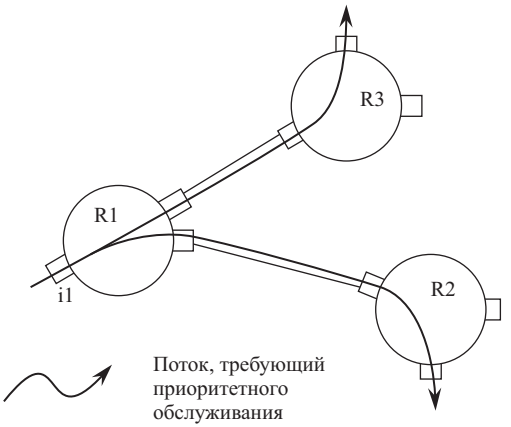


Рис. 2.1. В модели DiffServ объектами обслуживания являются классы трафика, а не потоки

модели, так как маршрутизаторы не должны запоминать состояния каждого пользовательского потока. Высокая степень масштабируемости Diffserv обеспечивается также тем, что каждый маршрутизатор самостоятельно принимает решение о том, как он должен обслуживать тот или иной класс трафика, не согласуя свои действия с другими маршрутизаторами. Такой подход назван *независимым поведением маршрутизаторов* (Per Hop Behavior, PHB). Так как в модели DiffServ маршруты пакетов не отслеживаются, то здесь не используется сигнальный протокол резервирования ресурсов, подобный протоколу RSVP в модели IntServ. Вместо этого маршрутизаторы сети выполняют статическое резервирование ресурсов для каждого из поддерживаемых сетью классов.

Требования к необходимому набору показателей качества обслуживания задаются в специальном однобайтовом поле каждого пакета — в октете Type of Service (ToS) протокола IPv4 или в октете Traffic Class (TC) протокола IPv6. Отметим, что в модели DiffServ это поле называется *DS-байтом*. Содержание DS-байта определяет вид предоставляемых услуг.

2.2. Обеспечение верности передачи данных

Защита от ошибок в системах без обратной связи. В системах без обратной связи (однонаправленных) для повышения верности приема используются следующие основные способы: многократная передача кодовых комбинаций; одновременная передача кодовой комбинации по нескольким параллельно работающим каналам; помехоустойчивое кодирование, т.е. использование кодов, исправляющих ошибки (корректирующих кодов).

Многократная передача кодовых комбинаций является наиболее просто реализуемым способом повышения верности. Повторение кодовых комбинаций может осуществляться вручную и автоматически (без участия операторов). Пусть передается буква А, число повторений возьмем равным пяти. Если на приемном конце имеем АБААС (буква А исказилась 2 раза, превратившись соответственно в Б и С), то выносится решение о том, что передавалась буква А, поскольку в последовательности из пяти букв она встречалась наиболее часто. Если в принятой последовательности ни одна из букв не повторяется, то принятое сообщение ликвидируется (стирается).

Главный недостаток такого способа — существенное уменьшение скорости передачи. В нашем примере скорость передачи информации уменьшается в 5 раз по сравнению со случаем однократной передачи кодовых комбинаций.

Способ повышения верности, основанный на снижении скорости передачи, широко применяется в технике передачи дискретных сообщений. Так, в системе ПД, построенной по рекомендации V.23, предусмотрены две скорости — 600 и 1200 Бод. Очевидно, что передача со скоростью 600 Бод равносильна передаче 2 раза подряд единичных элементов длительностью 1/1200 мс. Еще один пример. При переходе от использования модуляции 256 QAM к 16 QAM скорость передачи информации (бит/с) снижается в два раза, что позволяет повысить верность передачи.

При одновременной передаче кодовой комбинации по нескольким параллельным каналам (обычно число каналов нечетное) решение о том, какая кодовая комбинация передавалась, выносится методом голосования (т.е. так же, как и при многократной передаче кодовых комбинаций). Иногда передача осуществляется по двум параллельным каналам, и информация выбирается из того канала, качество которого в момент приема кодовой комбинации было наилучшим.

При передаче сообщений по N параллельным каналам скорость передачи информации не зависит от числа каналов. Однако при этом существенно возрастают (в N раз!) расходы на аренду каналов.

Более эффективно используются дискретные каналы при применении корректирующих кодов. В однонаправленных системах это должны быть коды, исправляющие ошибки. Широкое распространение на практике получили двоичные корректирующие коды, при формировании которых используются только два типа элементов: 0 и 1.

Построение корректирующих кодов. Каждому символу исходного алфавита сообщений объема N_a поставим в соответствие n-элементную двоичную последовательность — кодовую комбинацию. Возможное (общее) число последовательностей длины n составляет

$N_0 = 2^n$, причем должно соблюдаться условие $N_0 > N_a$.

Если $N_0 = N_a$, то все возможные последовательности n-элементного кода используются для передачи или, как говорят, являются разрешенными. Полученный таким образом код называется простым.

Пример 2.1. Для передачи сообщений, число которых равно восьми ($N_a = 8$), используется трехэлементный код. Число кодовых комбинаций, которое можно при этом получить, $N_0 = 2^3 = 8$. Из табл. 2.2 видно, что комбинация под номером 0 отличается от комбинации 1 только в одной позиции. Следовательно, если при передаче комбинации 000 произойдет ошибка в третьем элементе, то получим комбинацию 001.

Таблица 2.2. Кодовые комбинации трехэлементного кода

Номер комбинации	0	1	2	3	4	5	6	7
Вид комбинации	000	001	010	011	100	101	110	111

Степень различия комбинаций определяется *расстоянием Хемминга d*. Это расстояние для любых двух кодовых комбинаций определяется числом несовпадающих в них разрядов. Например, две ниже написанные друг под другом комбинации не совпадают в двух разрядах:

$$\begin{array}{r} \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array}\,0 \\ \oplus\quad 0\quad 0 \\ \hline 1\ 1\ 0 \end{array}$$

поэтому расстояние Хемминга $d = 2$. Иначе его определяют как вес суммы по модулю два ($\oplus$ — условное обозначение суммы) этих кодовых комбинаций. *Весом W* кодовой комбинации называется число входящих в нее ненулевых элементов.

Перебрав все возможные пары кодовых комбинаций, можно найти *минимальное* хеммингово *расстояние*, которое принято называть кодовым и обозначать d_0. Для примера 2.1 кодовое расстояние $d_0 = 1$. Рассмотренный в примере код простой. Любая ошибка (даже одиночная!) при использовании такого кода приведет к тому, что переданная разрешенная кодовая комбинация перейдет в другую разрешенную. Таким образом, простой код не способен обнаруживать и тем более исправлять ошибки и имеет $d_0 = 1$.

Для того, чтобы код мог обнаруживать ошибки, необходимо, чтобы соблюдалось неравенство $N_a < N_0$. При этом неиспользуемые n-элементные кодовые комбинации, число которых ($N_0 - N_a$), будем называть *запрещенными*. Они определяют избыточность кода. Очевидно, что появление ошибки в кодовой комбинации будет обнаружено, если переданная разрешенная комбинация перейдет в одну из запрещенных. В качестве $N_p = N_a$ разрешенных кодовых комбинаций надо

выбирать такие, которые максимально отличаются друг от друга.

Пример 2.2. Алфавит передаваемых сообщений $N_a = 2$. Выберем из числа комбинаций, представленных в табл. 2.2, две. Очевидно, что ими должны быть комбинации 000, 111 или 001 и 110 и т.д. Кодовое расстояние $d_0 = 3$. Ошибки кратности один или два превращают любую разрешенную кодовую комбинацию в запрещенную. Следовательно, максимальная кратность обнаруживаемых таким образом ошибок равна двум ($t_{\text{о.ош}} = 2$).

Нетрудно догадаться, что минимальное кодовое расстояние d_0 и гарантированно обнаруживаемая кратность ошибок связаны соотношением $t_{\text{о.ош}} = d_0 - 1$.

Исправление ошибок возможно также только в том случае, если переданная разрешенная кодовая комбинация переходит в запрещенную. Вывод о том, какая кодовая комбинация передавалась, делается на основании сравнения принятой запрещенной комбинации со всеми разрешенными. Принятая комбинация отождествляется с той из разрешенных, на которую она больше всего похожа, т.е. с той, от которой она отличается меньшим числом элементов. Так, если в примере 2.2 при передаче кодовой комбинации 000 получим 001, то вынесем решение, что передавалась кодовая комбинация 000.

Связь между d_0 и кратностью исправляемых ошибок определяется выражением $t_{\text{и.ош}} = \frac{d_0}{2} - 1$ для четного d_0 и $t_{\text{и.ош}} = \frac{d_0 - 1}{2}$ для нечетного d_0.

Итак, задача получения кода с заданной корректирующей способностью сводится к задаче выбора (путем перебора) из $N_0 = 2^n$ кодовых комбинаций N_a комбинаций с требуемым кодовым расстоянием d_0. Если n достаточно мало, то такой перебор не представляет особого труда. При больших n перебор может оказаться непосильным даже для современной ЭВМ, поэтому на практике используют методы построения кодов, не требующие перебора с целью получения кода с заданным d_0 и отличающиеся невысокой сложностью реализации.

Классификация корректирующих кодов. Помехоустойчивые или корректирующие коды (рис. 2.2) делятся на блочные и непрерывные.

К *блочным* относятся коды, в которых каждому символу алфавита сообщений соответствует блок (кодовая комбинация) из $n(i)$ элементов, где i — номер сообщения. Если $n(i) = n$, т.е. длина блока постоянна и не зависит от номера сообщения, то код называется *равномерным*. Такие коды чаще применяются на практике. Если длина блока зависит от номера сообщения, то блочный код называется *неравномерным*. Примером неравномерного кода служит код Морзе. В непрерывных кодах передаваемая информационная последователь-

ность не разделяется на блоки, а проверочные элементы[1] размещаются в определенном порядке между информационными.

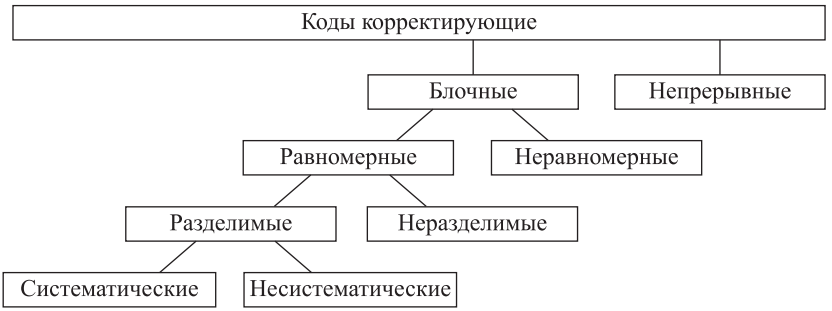


Рис. 2.2. Классификация корректирующих кодов

Равномерные блочные коды делятся на разделимые и неразделимые. В разделимых кодах элементы разделяются на информационные и проверочные, занимающие определенные места в кодовой комбинации, во-вторых, отсутствует деление элементов кодовых комбинаций на информационные и проверочные. К последним относится код с постоянным весом, например, рекомендованный Международным консультативным комитетом по телефонии и телеграфии (МККТТ), семиэлементный телеграфный код № 3 с весом каждой кодовой комбинации, равным трем.

Примерами систематических кодов являются коды Хемминга и циклические. Последние реализуются наиболее просто, что и привело к их широкому использованию в УЗО. Для систематического кода применяется обозначение (n, k)-код, где n — число элементов в комбинации; k — число информационных элементов.

Характерной особенностью этих кодов является также и то, что информационные и проверочные элементы связаны между собой зависимостями, описываемыми линейными уравнениями. Отсюда возникает и второе название систематических кодов — линейные.

Код Хемминга. Рассмотрим в качестве примера построение систематического кода с кодовым расстоянием $d_0 = 3$ (кода Хемминга). Пусть число сообщений, которое необходимо передать, равно 16. Тогда необходимое число информационных элементов $k = \log_2 N_a = 4$. Можно выписать все 16 кодовых комбинаций, включая нулевую (0000). Это один из возможных способов задания исходного (простого) кода. Другой способ заключается в выписывании только четырех

[1] Проверочные элементы в отличие от информационных, относящихся к исходной последовательности, служат для обнаружения и исправления ошибок и формируются по определенным правилам.

кодовых комбинаций простого кода в виде матрицы, называемой единичной:

$$\begin{Vmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{Vmatrix}. \qquad (2.1)$$

Суммируя по модулю два в различном сочетании кодовые комбинации, входящие в единичную матрицу, можно получить 15 кодовых комбинаций, 16-я — нулевая. Кодовые комбинации, составляющие матрицу (2.1), линейно независимы. Можно было бы составить матрицу и из других кодовых комбинаций (лишь бы они были линейно независимыми). Ненулевые комбинации A_1, A_2, A_3, A_4 линейно независимые, если $q_1A_1 \oplus q_2A_2 \oplus q_3A_3 \oplus q_4A_4 \neq 0$, где $q_i \in \{0,1\}$ при условии, что хотя бы один из коэффициентов $q_i \neq 0$. Дополним каждую кодовую комбинацию в (2.1) проверочными элементами так, чтобы обеспечивалось $d_0 = 3$. Будем иметь в виду также тот факт, что к числу разрешенных комбинаций корректирующего кода принадлежит и комбинация 0000...0, называемая *нулевой*. Очевидно, что в числе добавляемых проверочных элементов должно быть не менее двух единиц. Тогда общее число единиц в каждой комбинации кода получим не меньше трех, и комбинации, полученные нами, будут отличаться от нулевой, по крайней мере, в трех элементах. Добавим по две единицы к каждой строке матрицы (2.1):

$$\begin{Vmatrix} 1\,0\,0\,0\,1\,1 \\ 0\,1\,0\,0\,1\,1 \\ 0\,0\,1\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1 \end{Vmatrix}. \qquad (2.2)$$

Складывая строки 1 и 2 матрицы (2.2) по модулю два

$$\begin{array}{r} 1\,0\,0\,0\,1\,1 \\ \oplus\ 0\,1\,0\,0\,1\,1, \\ \hline 1\,1\,0\,0\,0\,0 \end{array}$$

видим, что они отличаются только в двух элементах, т.е. заданное кодовое расстояние не обеспечивается. Дополним каждую строку проверочными элементами так, чтобы $d_0 = 3$. Тогда матрица примет вид

$$\begin{Vmatrix} 1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,0 \end{Vmatrix}. \qquad (2.3)$$

Добавляемые проверочные элементы могут быть записаны и в другом порядке. Необходимо лишь обеспечить $d_0 = 3$.

Матрицу (2.3) называют *производящей*, или *порождающей*, матрицей кода (7,4), содержащего семь элементов, из которых четыре

информационных. Обычно матрицу обозначают буквой **G** с индексом, указывающим, к какому коду она относится (в нашем случае $\mathbf{G}_{(7,4)}$). Производящая матрица состоит из двух матриц — единичной (размерности $k \times k$) и $\mathbf{C}_{(r,k)}$, содержащей r столбцов и k строк. Суммируя в различном сочетании строки матрицы (2.3), получаем все (кроме нулевой) комбинации корректирующего кода с $d_0 = 3$.

Обозначим элементы комбинации полученного семиэлементного кода $a_1, a_2, a_3, a_4, a_5, a_6, a_7$, из которых a_1, a_2, a_3, a_4 — информационные и a_5, a_6, a_7 — проверочные. Последние могут быть получены путем суммирования по модулю два определенных информационных элементов. Разумеется, правило формирования проверочного элемента a_i для любой кодовой комбинации одинаково.

Найдем правило формирования элемента a_5, пользуясь матрицей (2.3). Из первой строки следует, что в суммировании должен обязательно участвовать элемент a_1 (только в этом случае $a_5 = 1$), из второй — что элемент a_3 в суммировании не должен участвовать, а из четвертой — что элемент a_4 должен участвовать в суммировании. Итак,

$$a_5 = a_1 \oplus a_2 \oplus a_4. \tag{2.4}$$

Уравнения для a_6 и a_7 по аналогии записываются в виде:

$$a_6 = a_1 \oplus a_3 \oplus a_4, \tag{2.5}$$

$$a_7 = a_1 \oplus a_2 \oplus a_3. \tag{2.6}$$

Алгоритм формирования проверочных элементов a_5, a_6, a_7 может быть задан матрицей, называемой *проверочной*. Эта матрица содержит r строк и n столбцов. Применительно к сформированному нами коду (7,4) она имеет вид:

$$\mathbf{B}_{(7,4)} = \left\| \begin{matrix} 1\,1\,0\,1\,1\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0 \\ 1\,1\,1\,0\,0\,0\,1 \end{matrix} \right\|.$$

Единицы, расположенные на местах, соответствующих информационным элементам матрицы $\mathbf{H}_{(7,4)}$, указывают на то, какие информационные элементы должны участвовать в формировании проверочного элемента. Единица на месте, соответствующем проверочному элементу, указывает, какой проверочный элемент получается при суммировании по модулю два информационных элементов. Так, из первой строки следует равенство

$$a_1 \oplus a_2 \oplus a_4 = a_5.$$

Процедура обнаружения ошибок основана на использовании проверок (2.4)–(2.6). Очевидно, что проверочные элементы, сформированные из принятых информационных, при отсутствии ошибок должны совпадать с принятыми проверочными.

Пример 2.3. Переданная кодовая комбинация имеет вид 1000111 (первая строка матрицы (2.3)). В результате действия помех на приемном конце имеем $a'_1, a'_2, a'_3, a'_4, a'_5, a'_6, a'_7 = 1100111$. Произведем проверки (2.4)–(2.6):

$$a'_1 \oplus a'_2 \oplus a'_4 = 1 \oplus 1 \oplus 0 = 0 = a^*_5, \tag{2.7}$$

$$a'_1 \oplus a'_3 \oplus a'_4 = 1 \oplus 0 \oplus 0 = 1 = a^*_6, \tag{2.8}$$

$$a'_1 \oplus a'_2 \oplus a'_3 = 1 \oplus 1 \oplus 0 = 0 = a^*_7. \tag{2.9}$$

В то же время $a'_5 = 1, a'_6 = 1, a'_7 = 1$, т.е. $a^*_5 \neq a'_5$, $a^*_7 \neq a'_7$, что говорит о наличии ошибок в принятой кодовой комбинации. При отсутствии в принятой кодовой комбинации ошибок $a'_5 \oplus a^*_5 = b_1 = 0$, $a'_6 \oplus a^*_6 = b_2 = 0$, $a'_7 \oplus a^*_7 = b_3 = 0$.

Комбинация $b_3 b_2 b_1$ называется *синдромом* (проверочным вектором). Равенство нулю всех элементов синдрома указывает на отсутствие ошибок или на то, что кодовая комбинация принята с ошибками, которые превратили ее в другую разрешенную. Последнее событие имеет существенно меньшую вероятность, чем первое.

Вид ненулевого синдрома определяется характером ошибок в кодовой комбинации. В нашем случае вид синдрома зависит от местоположения одиночной ошибки. В табл. 2.3 отражено соответствие между местоположением одиночной ошибки для кода, заданного матрицей (2.3), и видом синдрома.

Таблица 2.3. Местоположение ошибки и вид синдрома

Номер элемента, в котором произошла ошибка	1	2	3	4	5	6	7
Вид синдрома	111	101	110	011	001	010	100

Таким образом, зная вид синдрома, можно определить место, где произошла ошибка, и исправить принятый элемент на противоположный.

Пример 2.4. Передавалась кодовая комбинация 1000111. Принята кодовая комбинация 0000111. Синдром имеет вид 111. В соответствии с табл. 2.3, исказился первый элемент (a_1). Изменим первый элемент на противоположный:

$$\begin{array}{r} 0\,0\,0\,0\,1\,1\,1 \\ \oplus\ 1\,0\,0\,0\,0\,0\,0 \\ \hline 1\,0\,0\,0\,1\,1\,1 \end{array}.$$

Полученная в результате исправления ошибки кодовая комбинация совпадает с переданной.

Рассмотренный код (7,4) гарантированно обнаруживает двукратные ошибки, а исправляет только однократные ошибки.

Циклические коды. В теории циклических кодов кодовые комбинации обычно представляются в виде полинома. Так, n-элементная кодовая комбинация записывается в виде

$$A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0,$$

где $a_i = \{0, 1\}$, причем $a_i = 0$ соответствуют нулевым элементам комбинации, а $a_i = 1$ — ненулевым. Например, комбинациям 1101 и 1010 соответствуют многочлены $A_1(x) = x^3 + x^2 + 1$ и $A_2(x) = x^3 + x$.

При формировании комбинаций циклического кода часто используют операции сложения многочленов и деления одного многочлена на другой. Так,

$$A_1(x) + A_2(x) = (x^3 + x^2 + 1) + (x^3 + x) = x^2 + x + 1,$$

поскольку $x^3 + x^3 = x^3(1 \oplus 1) = 0$.

Рассмотрим операцию деления на следующем примере:

$$\begin{array}{l|l} x^7 + x^5 + x^4 + x^2 + 1 & x^3 + x^2 + 1 \\ \underline{x^7 + x^6 + x^4} & \overline{x^4 + x^3 + 1} \\ x^6 + x^5 + x^2 & \\ \underline{x^6 + x^5 + x^3} & \\ x^3 + x^2 + 1 & \\ \underline{x^3 + x^2 + 1} & \\ 0 \quad 0 \quad 0 & \end{array}$$

Деление выполняется как обычно, только вычитание заменяется суммированием по модулю два.

Разрешенные комбинации циклического кода обладают двумя очень важными отличительными признаками: циклический сдвиг разрешенной комбинации тоже приводит к разрешенной кодовой комбинации. Все разрешенные кодовые комбинации делятся без остатка на полином $P(x)$, называемый *образующим*. Эти свойства используются при построении кодов, кодирующих и декодирующих устройств, а также при обнаружении и исправлении ошибок.

Найдем алгоритмы построения циклического кода, удовлетворяющего перечисленным выше условиям. Задан полином

$$P(x) = a_{r-1}x^r + a_{r-2}x^{r-1} + \ldots + 1,$$

определяющий корректирующую способность кода, и задан исходный простой код, который требуется преобразовать в корректирующий циклический.

Обозначим через $Q(x)$ многочлен, соответствующий комбинации простого кода. Возьмем произведение $Q(x)x^r$ и разделим его на $P(x)$. В результате получим многочлен $G(x)$ и остаток $R(x)/P(x)$:

$$\frac{Q(x)x^r}{P(x)} = G(x) + \frac{R(x)}{P(x)}. \qquad (2.10)$$

Умножим левую и правую части на $P(x)$, тогда (2.10) перепишет-

ся в виде

$$Q(x)x^r = G(x)P(x) + R(x). \qquad (2.11)$$

Перепишем равенство (2.11) в виде

$$G(x)P(x) = Q(x)x^r + R(x). \qquad (2.12)$$

Левая часть (2.12) делится без остатка на $P(x)$, значит, без остатка делится и правая часть. Из (2.12) вытекают два способа формирования комбинаций циклического кода: путем умножения многочлена $G(x)$ на $P(x)$ и путем деления $Q(x)x^r$ на $P(x)$ и приписывания к $Q(x)x^r$ остатка от деления $R(x)$.

Пример 2.5. Задан полином $G(x) = x^3 + x$, соответствующий комбинации простого кода. Сформировать комбинацию циклического кода (7,4) с производящим полиномом $P(x) = x^3 + x^2 + 1$. Можно получить комбинацию циклического кода в виде

$$G(x)P(x) = (x^3 + x)(x^3 + x^2 + 1) = x^6 + x^5 + x^4 + x.$$

Однако в полученной комбинации нельзя отделить информационные элементы от проверочных, и код получается неразделимым.

Перейдем ко второму способу, который чаще всего применяется на практике. Проделаем необходимые операции по получению комбинации циклического кода:

1) $G(x)x^r = (x^3 + x)x^3 = x^6 + x^4$;

2)
$$\begin{array}{l|l} x^6 + x^4 & x^3 + x^2 + 1 \\ \hline x^6 + x^5 + x^3 & x^3 + x^2 + 1 \\ \hline x^5 + x^4 + x^3 & \\ \hline x^5 + x^4 + x^2 & \\ \hline x^3 + x^2 & \\ \hline x^3 + x^2 + 1 & \\ \hline R(x) = 1 & \end{array}$$

3) $(x^6 + x^4 + 1)$ — комбинация циклического кода, полученная методом деления на производящий полином. Она может быть переписана в виде 1010001. Первые четыре элемента — информационные, последние три — проверочные, т.е. полученный код — разделимый.

Для обнаружения ошибок в принятой кодовой комбинации достаточно поделить ее на производящий полином. Если принятая комбинация разрешенная, то остаток от деления будет нулевым. Ненулевой остаток свидетельствует о том, что принятая комбинация содержит ошибки. По виду остатка (синдрома) можно в некоторых случаях также сделать вывод о характере ошибки и исправить ее.

Циклические коды достаточно просты в реализации, обладают высокой корректирующей способностью (способностью исправлять и обнаруживать ошибки) и поэтому рекомендованы МСЭ-Т для применения в аппаратуре ПД. Согласно рекомендации V.41, в системах

ПД с ОС рекомендуется применять код с производящим полиномом $P(x) = x^{16} + x^{12} + x^5 + 1$.

Эффективность применения корректирующих кодов. Полезный эффект от применения корректирующих кодов заключается в повышении верности. Вероятность неправильного приема кодовой комбинации простого кода определяется как вероятность появления в кодовой комбинации хотя бы одной ошибки, т.е.

$$P_{\text{ош}}^{(\text{п})} = 1 - (1 - p_{\text{ош}})^k,$$

где $p_{\text{ош}}$ — вероятность неправильного приема единичного элемента; k — число элементов в комбинации простого кода. При применении систематических корректирующих кодов к исходной кодовой комбинации добавляются проверочные элементы, позволяющие исправлять или обнаруживать ошибки. Так, если код используется в режиме исправления ошибок и кратность исправляемых ошибок $t_{\text{и.ош}}$, то вероятность неправильного приема кодовой комбинации

$$P_{\text{ош}}^{(\text{и})} = \sum_{t=t_{\text{и.ош}}+1}^{n} C_n^t p_{\text{ош}} (1 - p_{\text{ош}})^{n-t}.$$

В результате применения корректирующего кода в режиме исправления ошибок вероятность ошибки уменьшается в $K_{\text{и}}$ раз: $K_{\text{и}} = P_{\text{ош}}^{(\text{п})} / P_{\text{ош}}^{(\text{и})} > 1$. Однако это достигается за счет увеличения затрат на реализацию системы и снижения скорости передачи информации. Если в системе с простым кодом скорость равна $C_{\text{п}}$, то в системе с корректирующим кодом скорость $C_{\text{к}} = C_{\text{п}} \cdot \gamma_1$, где $\gamma_1 = k/n$ — коэффициент, характеризующий потери скорости вследствие введенной в код избыточности. Чем больше избыточность (меньше γ_1), тем меньше скорость передачи информации, т.е. тем меньше в единицу времени передается полезной информации.

Качество реальных каналов во времени меняется, и если заданы требования на верность передачи, то необходимо ввести такую избыточность, которая обеспечивала бы заданную верность даже при самом плохом качестве канала. Напрашивается мысль о целесообразности изменения избыточности, вводимой в кодовую комбинацию, по мере изменения характеристик канала связи. Системы, в которых меняется избыточность с изменением качества канала, относятся к числу адаптивных. Одним из типов адаптивных систем являются системы с обратной связью. В этих системах между приемником и передатчиком помимо основного (прямого) канала имеется вспомогательный (обратный).

Следует заметить, что системы без ОС используются обычно только тогда, когда нельзя организовать канал обратной связи или когда предъявляются жесткие требования к времени задержки сооб-

щения. *Временем задержки* кодовой комбинации называется время от момента выдачи ее первого элемента источником сообщений до момента получения последнего элемента комбинации получателем сообщений.

Системы с обратной связью характеризуются повторением кодовых комбинаций, в которых обнаружены ошибки. Решение о необходимости повторения может выноситься на приеме (системы с решающей обратной связью — РОС) или на передаче (системы с информационной обратной связью — ИОС).

Как уже отмечалось, системы с обратной связью отличаются наличием канала, по которому осуществляется «служебная» связь передатчика с приемником. В системах с РОС приемником определяется наличие в принятой комбинации ошибки или вычисляется вероятность того, что кодовая комбинация содержит ошибки. Если в кодовой комбинации обнаружены ошибки или вероятность того, что в ней содержатся ошибки, оказалась достаточно большой, то по обратному каналу посылается сигнал решения о необходимости повторения (отсюда название *решающая обратная связь*).

Соответствующий аналог передачи с РОС можно найти и в телефонной связи. Если вследствие действия помех не расслышано слово, то обычно просят его повторить.

В системах с ИОС принятая кодовая комбинация A_i^* возвращается на передающую сторону по обратному каналу, где она сравнивается с переданной комбинацией A_i. Последнюю можно рассматривать как эталонную комбинацию. Если комбинации A_i^* и A_i различаются, то комбинация передается повторно. При разговоре по телефону также часто используют ИОС, когда в условиях сильных помех просят собеседника повторить переданное ему сообщение, чтобы убедиться, что он его воспринял правильно.

Системы с РОС получили наибольшее практическое распространение. Существуют различные разновидности этих систем. Простейший и довольно часто применяемый на практике алгоритм работы системы с РОС заключается в следующем. Источник сообщений ИС выдает в кодер (рис. 2.3) первую кодовую комбинацию (или блок, состоящий из нескольких кодовых комбинаций). К исходным элементам в кодере добавляются проверочные. Комбинация выдается в дискретный канал и одновременно записывается в накопитель $Н_1$ (накопитель передачи). После выдачи первой кодовой комбинации источник ждет ответа о том, как она принята.

Принятая кодовая комбинация декодируется. Информационные элементы записываются в накопитель приема ($Н_2$). Если ошибка не обнаружена, то по команде управляющего устройства информацион-

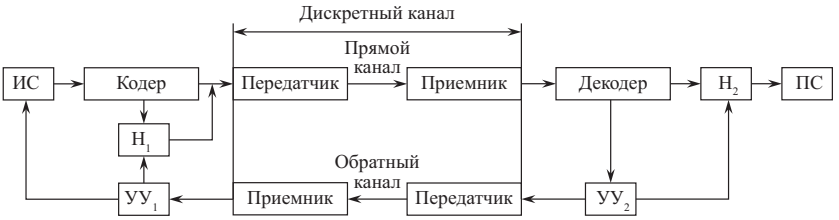


Рис. 2.3. Функциональная схема системы с РОС-ОЖ

ные элементы из накопителя выдаются получателю, а по обратному каналу выдается сигнал «Да», подтверждающий правильность приема переданной кодовой комбинации номер один (обратный канал будем пока считать идеальным). По сигналу «Да» управляющее устройство стирает из $Н_1$ кодовую комбинацию и дает разрешение на выдачу от источника следующей кодовой комбинации. Если следующая комбинация исказилась и ошибки на приеме обнаружены, то по команде $УУ_2$ информация из $Н_2$ стирается, а по обратному каналу выдается сигнал «Нет». По этому сигналу на передающем конце $УУ_1$ запрещает выдачу следующей кодовой комбинации от источника и дает команду о повторной выдаче искаженной комбинации из накопителя $Н_1$. Теоретически кодовая комбинация может повторяться бесконечное число раз. Обычно после определенного числа повторений (например, трех) кодовая комбинация стирается. Очевидно, что чем больше повторений на анализируемом интервале времени, тем хуже качество канала, тем дольше длится «перекачка» сообщения от источника и тем ниже скорость передачи информации.

Рассматриваемый алгоритм работы системы называется алгоритмом с ожиданием, а сама система передачи дискретных сообщений — системой с решающей обратной связью и ожиданием (РОС-ОЖ). Такие системы довольно часто используются для передачи дискретных сообщений. Основное их достоинство — простая техническая реализация. К недостаткам следует отнести существенные потери скорости передачи информации, источником которых, помимо введенных в кодовую комбинацию проверочных элементов и переспросов, являются потери на ожидание ответа со стороны приемника. При этом скорость передачи информации определяется выражением

$$C = B\gamma_1\gamma_2\gamma_3, \tag{2.13}$$

где $\gamma_1, \gamma_2, \gamma_3$ — соответственно коэффициенты, характеризующие потери скорости, обусловленные наличием в кодовой комбинации проверочных элементов; ожиданием сигнала решения о качестве приема; повторными передачами кодовых комбинаций. Очевидно, что процент потерь скорости определяется как $(1-\gamma_1)\cdot 100\%$.

Учитывая, что время, необходимое для передачи информацион-

ных элементов одной кодовой комбинации, равно $k\tau_0$, а время, затрачиваемое на передачу одной кодовой комбинации при однократной передаче, равно $n\tau_0 + t_{ож}$, где $t_{ож}$ — время ожидания сигнала решения (время от момента передачи в канал одной кодовой комбинации до момента передачи следующей), получаем

$$\gamma_1\gamma_2 = \frac{k\tau_0}{n\tau_0 + t_{ож}} = \frac{k}{n} \cdot \frac{1}{1 + \frac{t_{ож}}{n\tau_0}}.$$

Таким образом,

$$\gamma_2 = \frac{1}{1 + \frac{t_{ож}}{n\tau_0}}.$$

Следовательно, потери на ожидание будут тем меньше, чем меньше скорость модуляции (больше τ_0) или при данной скорости модуляции больше длина кодовой комбинации. Коэффициент γ_3 в (2.13) есть величина, определяемая как $(1 - P_{оо})$, где $P_{оо}$ — вероятность обнаружения в кодовой комбинации ошибок. Чем больше длина кодовой комбинации, тем больше $P_{оо}$ и меньше γ_3. Нетрудно догадаться, что из этого следует возможность оптимизации скорости путем изменения длины кодовой комбинации.

В системах с РОС и непрерывной передачей информации отсутствуют потери на ожидание ($t_{ож} = 0,\ \gamma_2 = 1$). В этих системах при обнаружении ошибок в принятой кодовой комбинации производится повторение этой комбинации и ряда других, примыкающих к ней. Для уменьшения потерь на переспросы иногда по каналу обратной связи передается адрес (номер) кодовой комбинации, которую надо повторить. Такой метод применяется в системах с РОС и адресным переспросом. Однако непрерывная передача информации и тем более адресный переспрос требуют существенного усложнения аппаратуры ПД, что, в свою очередь, приводит к ее удорожанию и снижению надежности.

В простейших системах с ИОС для передачи информации по прямому каналу можно использовать простые коды (без избыточности), и тогда обратный канал должен иметь такую же пропускную способность, что и прямой. В системах с РОС любого типа по обратному каналу передаются только сигналы решения и обратный канал имеет существенно меньшую пропускную способность. Так, при передаче информации со скоростью 600/1200 Бод по прямому каналу в обратном узкополосном канале передача осуществляется со скоростью не более чем 75 Бод. Возможность использования узкополосного канала в качестве обратного — существенное преимущество систем с РОС, делающее их применение на практике более предпочтительным по сравнению с системами с ИОС.

2.3. Обеспечение показателей структурной надежности

Резервирование и динамическое восстановление. Отказы в телекоммуникационной сети вызывают:

- обрыв установленных соединений;
- блокировку вызовов вследствие ограниченности ресурсов;
- рост числа вызовов за счет повторных вызовов со стороны пользователей, утративших соединение;
- потерю доверия со стороны пользователей сети.

Исходя из необходимости выполнения SLA, операторы вынуждены закладывать в свой бюджет затраты на обеспечение требуемой структурной надежности и компенсацию финансовых претензий потребителей при нарушении условий соглашения SLA.

Процесс восстановления связи между двумя конечными узлами может происходить путем перенаправления трафика на заранее подготовленный до установления соединения резервный путь (proactive). Этот метод принято называть резервированием (reservation) или защитой переключением. Другим вариантом восстановления соединения является поиск нового пути (перемаршрутизация) после возникновения отказа. Этот метод принято называть восстановлением (restoration) или динамическим (reactive) восстановлением. Последний термин является, на наш взгляд, более предпочтительным, так как переход на заранее подготовленный резервный путь тоже является восстановлением соединения. Однако, для сокращения вместо термина «динамическое восстановление» будем пользоваться в дальнейшем термином «восстановление». Кстати, в англоязычной литературе используется именно этот термин (restoration).

Достоинством метода резервирования является быстрое восстановление связи, недостатком — необходимость в дополнительной, иногда очень существенной, пропускной способности для организации резервного пути. Метод восстановления требует больших затрат времени на восстановление связи, кроме того, возникает риск нестабильности сети, особенно в случае частых самоустраняющихся отказов [5]. Достоинством метода восстановления является лучшее использование пропускной способности сети связи. Описание и анализ методов резервирования можно, например, найти в статьях [6–11], методов восстановления — в [12–15], а также монографиях [16, 17].

Резервирование и восстановление позволяют обеспечить требуемый потребителем показатель готовности соединения или показатель готовности услуги.

Готовность — это вероятность того, что соединение будет обеспечено в любой случайный момент. Это важнейшая метрика, отражающая структурную надежность сети.

При выборе сетевого сервиса, готовность предоставления услуги является одной из многих тесно связанных метрик, иногда даже более важных, чем другие QoS-параметры — такие как задержка, джиттер, потеря пакетов.

Анализ рынка телекоммуникационных услуг показывает, что 50% пользователей услуг ожидают, по крайней мере, 99,9% доступности сервиса [18]. Финансовые потери в результате отсутствия связи на бирже в течение 1 минуты чреваты убытками порядка 110 000$ [18]. Поэтому для бизнес-клиентов требуется обеспечить коэффициент готовности 0,999999, что соответствует длительности простоя в год 0,53 минуты или шестому классу готовности (табл. 2.4).

Таблица 2.4. Классы готовности систем

Тип системы	Недоступность (мин/год)	Доступность	Класс готовности
Необслуживаемые	50 000	90%	1
Обслуживаемые	5 000	99%	2
Хорошо обслуживаемые	500	99,9%	3
Отказоустойчивые	50	99,99%	4
Высокая готовность	5	99,999%	5
Очень высокая готовность	0,5	99,9999%	6
Сверхвысокая готовность	0,05	99,99999%	7

Заметим, что более высокая готовность требует более высоких затрат со стороны оператора, что сказывается на цене услуги. А так как разные пользователи имеют различную чувствительность к доступности сервиса, то пользователи с лимитированным бюджетом должны иметь удовлетворяющий их по цене уровень доступности. Отсюда возникла идея обеспечения эластичного (гибкого) доступа к сервису, которая предполагает, в частности, разные варианты резервирования в зависимости от требований потребителя к коэффициенту готовности. Более того, сопоставляя финансовые потери при отказах и затраты на обеспечение $K_г$ можно найти наиболее приемлемое, с точки зрения потребителя, значение $K_г$ [19].

Надежность функционирования сетевой инфраструктуры обеспечивается путем использования алгоритмов резервирования и восстановления связи между сетевыми узлами и средств повышения надежности самих узлов, в первую очередь маршрутизаторов и коммутаторов. Сегодня все серьезные технические решения требуют как минимум двух модулей управления, характеризуются избыточностью различных подсистем с возможностью их быстрой замены в «горячем» режиме.

Для определения степени защиты, требуемой для данного участ-

ка сети, необходимо учитывать вероятность отказа участка сети и предполагаемые воздействия на трафик (в понятиях времени восстановления, вероятности потери пакетов) [7].

Значение вероятности отказа заданного участка сети (то есть области защиты) можно определить на основании доступной информации о характере происходящих отказов. Начальное значение вероятности отказа может быть уточнено на основе фактической статистики отказов.

Если вероятность отказа известна, необходимо рассмотреть, как отказ влияет на трафик в сети, то есть определить «степень воздействия отказа». Критическим аспектом для оценки воздействия отказа является гарантируемое качество обслуживания (QoS) трафика, которое определяется двумя компонентами: временем восстановления и количеством потерянных пакетов.

Время восстановления $T_в$ определяется циклом восстановления пути. Заметим, что этот цикл можно упрощенно задать следующими составляющими:

1) временем обнаружения отказа T_1;
2) временем удержания (в случае необходимости) T_2;
3) временем уведомления (т.е. посылки сообщения узлу, ответственному за переключение) T_3;
4) временем для резервирования маршрута и сигнализации T_4;
5) временем для переключения трафика T_5 с активного пути на резервный путь.

Зная время восстановления $T_в$ и скорость передачи пакетов R, можно определить количество потерянных пакетов $N_{пп} = RT_в$.

Сокращение времени обнаружения отказа и времени переключения зависит от используемой технологии восстановления. Кроме того, время установления резервных путей (при обнаружении отказа) зависит от метода маршрутизации и используемых методов сигнализации.

Сокращение времени уведомления $T_{ув}$ — вероятно, основной аспект при проектировании методов защиты для сети. Время уведомления зависит от времени распространения между узлами сигнала об отказе $T_р$ и от расстояния $D(i, a)$, которое может быть определено как количество участков сети (ребер) между узлом, обнаружившим отказ (узел a), и узлом, ответственным за переключение (узел i).

$$T_{ув} = T_р \cdot D(i, a).$$

Так как время распространения сигнала об отказе зависит от характера среды распространения сигнала, то снижение может быть достигнуто уменьшением расстояния ($D(i, a)$). Местное (local) резервирование обеспечивает оптимальное значение ($D(i, a) = 0$). Главная проблема состоит в том, что расстояние $D(i, a)$ неизвестно заранее,

потому что неизвестно, какое звено пути выйдет из строя. Однако знание вероятностей отказа участков пути можно использовать, чтобы оценить вероятности появления тех или иных значений $D(i,a)$ и вычислить среднее значение $D(i,a)$

$$\overline{D}(i,a)=\sum_i P_i D(i,a).$$

При проектировании сети необходимо стремиться уменьшить как вероятность отказа, так и степень воздействия отказа на качество предоставляемых услуг. Это непростая задача, поскольку существует взаимная связь между снижением степени воздействия отказа и снижением вероятности отказа.

Современные телекоммуникационные сети — это сети, обладающие огромной пропускной способностью и использующие на физическом уровне, как правило, волоконно-оптические линии связи. Поэтому задача обеспечения структурной надежности таких сетей является чрезвычайно актуальной.

Модели резервирования. Рассмотрим классические модели резервирования сетей связи (рис. 2.4) и дадим их краткую характеристику. Физическая топология сети состоит из узлов, соединенных линиями связи (каналами связи, звеньями). При рассмотрении процесса передачи от источника к получателю вводится понятие «путь». Различают первичные (рабочие) пути и пути резервные. Отрезок пути, состоящий из нескольких звеньев, принято называть «сегментом». Понятие «сегмента» можно рассматривать как обобщение понятий «рабочий путь» и «звено».

На рис. 2.4,а представлена модель защиты звена. Здесь каждое звено защищается в индивидуальном порядке (локальная защита). Этот метод обладает высокой вычислительной эффективностью, обеспечивает быструю перемаршрутизацию, прост и масштабируем, способен обеспечить устойчивость к множественным отказам, но нуждается в больших сетевых ресурсах.

Защита пути (рис. 2.4,б) осуществляется из конца в конец, т.е. от источника до получателя (иногда такую защиту называют глобальной). Здесь сетевые ресурсы используются более экономно, но вычисление пути из конца в конец является более сложной задачей. Различают два варианта такой защиты: а) альтернативный путь, использующий одно или несколько звеньев рабочего пути; б) альтернативный путь, ни в одном из звеньев не совпадающий с первичным путем. Второй вариант представляет интерес для случая, когда отказы могут произойти в любом из звеньев первичного пути (тогда как для каждого из случаев отказа в первом варианте пришлось бы искать свой альтернативный путь). Восстановление при втором варианте мо-

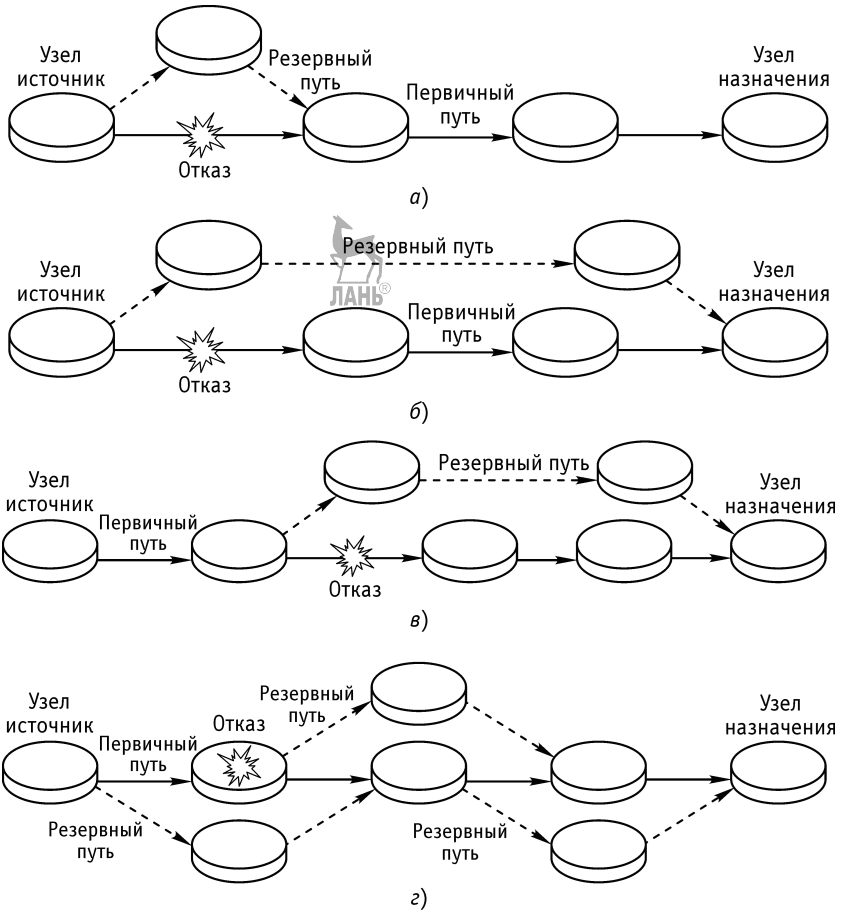


Рис. 2.4. Классические модели резервирования: *а* — защита звена; *б* — защита пути; *в* — защита сегмента; *г* — защита узла

жет быть начато немедленно после обнаружения отказа любого звена первичного пути без ожидания конкретизации звена, вышедшего из строя.

На рис. 2.4,в представлена модель защиты сегмента (участка из нескольких звеньев).

На примере моделей, представленных на рис. 2.4, мы рассмотрели только общие подходы к выбору области защиты (или масштаба защиты). Ниже мы остановимся на известных методах использования ресурсов пропускной способности, таких как $1+1$, $1:1$, $M:N$, которые могут использоваться как в вариантах защиты пути, так и звена или сегмента.

Защита пути 1 + 1. Данные передаются одновременно по рабочему и резервному пути. На приеме выделяется лучший сигнал. Рабочий и резервный пути разделены.

Защита звена 1 + 1. Принцип действия такой же, как в случае защиты пути, но обеспечивается обход только одиночного отказавшего звена или узла, а не всего пути.

Защита звена 1 : 1. До отказа данные посылаются только по рабочему пути. Второстепенный трафик может транслироваться по резервному пути. В случае отказа на рабочем звене прекращается передача второстепенного трафика и данные передаются по резервному звену, который становится рабочим.

В случае устранения отказа рабочего пути возможны следующие варианты:

1) трафик с резервного пути перебрасывается обратно на рабочий путь;
2) трафик после устранения отказа остается на резервном пути, в то время как рабочий путь выполняет функцию резервного.

Достоинством первого варианта является использование приоритетным трафиком более надежного пути, каковым обычно является рабочий путь. Недостатком — необходимость переключения, которое выполняется устройством, имеющим $K_г < 1$.

Защита пути 1 : 1. Принцип действия такой же, как в случае защиты звена 1 : 1, но здесь обеспечивается обход всего поврежденного пути. В вариантах 1 + 1 и 1 : 1 использовались выделенные ресурсы пропускной способности, в качестве которых можно рассматривать, в том числе и волокно, и оптический канал.

Защита звена $M : N$ (M — число резервных звеньев, N — число рабочих звеньев, $M < N$). Это так называемая групповая защита (shared). В случае отказа на рабочем звене, данные переключаются на резервное звено, но если число поврежденных рабочих звеньев превышает M, трафик теряется. Наиболее часто используемый вариант $M : N$ соответствует случаю, когда $M = 1$ $(1 : N)$. Применительно к MPLS защита звена $M : N$ имеет название «быстрая перемаршрутизация» (fast reroute) [12].

Защита пути $M : N$ ($M < N$). Принцип действия такой же, как и в случае защиты звена, но здесь обеспечивается обход всего пути. Этот метод резервирования наиболее востребован вследствие своей низкой стоимости и гибкости. Однако он достаточно сложен в оптимизации, особенно в случае, если используются механизмы, учитывающие приоритеты.

Рассмотренные выше процедуры резервирования могут использоваться совместно с процедурами восстановления. Так, например,

можно разделить весь трафик на несколько типов в соответствии с их приоритетами и, соответственно, разной чувствительностью к времени восстановления соединения. Для наиболее чувствительного к задержкам трафика можно применить защиту 1+1 или 1:1, для менее чувствительных — алгоритмы восстановления. Другой вариант совмещения — перейти при одновременном сбое на резервном и первичном (рабочем) каналах в схеме 1+1 на использование одного из возможных алгоритмов восстановления. Разумеется, перечислением этих двух вариантов перечень возможных подходов к проблеме совместного использования процедур резервирования и восстановления не исчерпывается.

Обобщенная характеристика методов обеспечения структурной надежности в условиях отказов представлена в табл. 2.5

Заметим, что классификация, представленная в табл. 2.5, может быть расширена с учетом следующих соображений. Механизм восстановления может работать между двумя узлами в одном домене (intradomain) или между двумя узлами разных доменов (interdomain) (рис. 2.5). Кроме того, учитывая, что большинство сетей имеют несколько уровней (например, IP/ATM/SDH/WDM), механизм восстановления может охватывать один (Single Layer) или сразу несколько уровней (Multiple Layers). При решении задачи обеспечения структурной надежности в многоуровневых сетях приходится решать различного рода вопросы: какой уровень (или какие несколько уровней) следует задействовать для обеспечения показателей структурной надежности? Если задействовано несколько уровней, то как обеспечить согласование между ними? Оптимальное решение подразумевает обеспечение компромисса между доступностью сетевых ресурсов, временем восстановления, объемом использованных сетевых ресурсов, сложностью реализации алгоритмов восстановления и т.п.

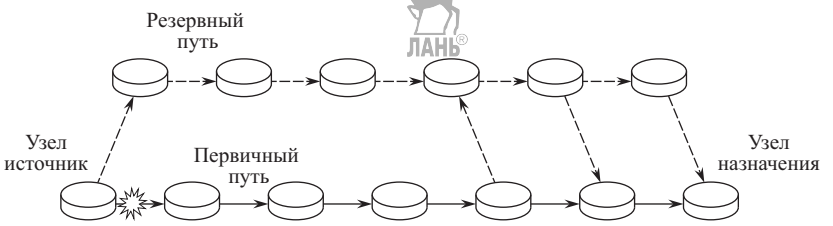


Рис. 2.5. Защита домена

В сетях IP/MPLS поверх DWDM можно вводить механизмы обеспечения отказоустойчивости только на нижнем физическом уровне, только на верхнем уровне, можно рассматривать различные комбинации использования нижних и верхних уровней [19].

Таблица 2.5. Опции защиты

Методы обеспечения структурной надежности

Защитное переключение (резервирование)	Восстановление (перемаршрутизация)

Выделение ресурсов

Предварительное	По требованию

Использование ресурсов

Выделенные	Общие	Второстепенного трафика

Создание пути

Предварительное	В соответствии с требуемым качеством	По требованию

Масштаб защиты

Глобальная (пути)	Сегмента

Защитное переключение

Автоматическое (внутренний сигнал)	Внешние команды

Опираясь на материал, опубликованный в [19], дадим обобщенную сравнительную характеристику представленных выше вариантов защиты:

1) общая защита $(1:N)$ более чувствительна к множественным отказам, чем индивидуальная $(1:1,\ 1+1)$;
2) защита пути более чувствительна к множественным отказам, чем индивидуальная;
3) общая защита пути позволяет более эффективно использовать доступную пропускную способность, чем индивидуальная;
4) защита пути позволяет более эффективно использовать доступную пропускную способность, чем защита звена;
5) восстановление пути отличается большей эффективностью восстановления (restoration efficiency), чем восстановление звена. В то же время восстановление звена происходит быстрее, чем восстановление пути.

Структурная надежность сетей зависит не только от надежности путей, звеньев или сегментов пути, но и от надежности узлов, т.е. от аппаратной надежности. Обеспечение последней может достигаться за счет резервирования. Резервирование вызывает увеличение габаритов, массы и потребляемой мощности, усложняется проверка аппаратуры и ее обслуживание, увеличивается себестоимость.

Основным параметром резервирования является кратность резервирования — отношение числа резервных устройств к числу рабочих устройств. Здесь также различают общее и раздельное резервирование. Однако под общим резервированием узла обычно понимают ре-

зервирование узла в целом; раздельное резервирование предполагает разбиение элементов узла на подсистемы, каждая из которых резервируется отдельно. В зависимости от способа подключения резервного узла различают горячее и холодное резервирование. При горячем резервировании резервный узел (узлы) работает в одинаковых условиях с основным и выполняет все функции основного. При холодном резервировании резервный узел работает в облегченных условиях. В этом случае он включается в работу только в случае выхода из строя рабочего узла.

Итак, обеспечение необходимой отказоустойчивости требует, как правило, внесения в систему или отдельно взятый ее компонент некоторой избыточности. Ее наличие в структуре системы обеспечивает возможности сверх тех, что могут быть обеспечены при функционировании в отсутствие отказов.

Применительно к узлам можно говорить об аппаратной избыточности (резервирование), некоторые особенности которой были рассмотрены выше. Существуют также такие виды избыточности, как программная, временная и информационная избыточность.

Программная избыточность (Software Redundancy) используется для контроля и обеспечения достоверности наиболее важных решений по управлению и обработке информации. Она заключается в сопоставлении результатов обработки одинаковых исходных данных разными программами и исключении искажения результатов, обусловленных различными аномалиями.

Временная избыточность (Time Redundancy) заключается в использовании некоторой части производительности компьютера для контроля за исполнением программ и восстановления (рестарта) вычислительного процесса (запас времени для повторного выполнения операции, например, двойного, тройного просчета).

Примером информационной избыточности является избыточная пропускная способность. Отсутствие точных методов расчета необходимой пропускной способности и быстрый рост трафика передачи данных вынуждает проектировать сети с очень высокой по пропускной способности избыточности.

2.4. QoS маршрутизация

Для многих приложений реального времени фундаментальной задачей является нахождение подходящего пути, который должен удовлетворять множеству ограничений. Эта задача (проблема) известна как задача QoS маршрутизации (выбора пути) с множеством ограничений MCP. В англоязычной литературе аббревиатура MCP расшифровывается как Multi-Constrained Path или Multi-Constrained Problem, т.е. путь с многими ограничениями или проблема множества

ограничений. В качестве таких ограничений могут выступать требования к значениям коэффициента готовности, задержке, вероятности потери пакетов, коэффициенту ошибок, стоимости мероприятий по обеспечению QoS и т.п.

При решении такого рода задач каждое звено пути характеризуется множеством весовых показателей (метрик).

Метрики подразделяются на классы. Класс аддитивных метрик (весов) характеризуется тем, что веса звеньев, составляющих путь (P), складываются

$$\omega_p = \sum_{i=1}^{N} \omega_i,$$

где ω_i — вес i-го звена, N — число звеньев.

Класс мультипликативных метрик отличается тем, что веса звеньев, составляющих путь, перемножаются

$$\omega_p = \prod_{i=1}^{N} \omega_i. \qquad (2.14)$$

Перейти от мультипликативной метрики к аддитивной можно путем логарифмирования (2.14)

$$\log \omega_p = \sum_{i=1}^{N} \log \omega_i. \qquad (2.15)$$

Примером аддитивной метрики является задержка, которая для пути вычисляется как сумма задержек звеньев. Примером мультипликативной метрики является коэффициент готовности. Аддитивная и мультипликативная метрики могут быть отнесены к кумулятивным метрикам от латинского Cumulo (накопление).

В качестве особого класса метрик можно выделить некумулятивные метрики. Возьмем пропускную способность пути, состоящего из N звеньев. Очевидно, что пропускная способность пути будет определяться звеном, имеющим наименьшую пропускную способность:

$$\omega_p = \min \omega_i, \quad i = 1, 2, ..., N, \qquad (2.16)$$

где ω_i — пропускная способность i-го звена.

В англоязычной литературе применительно к этому классу метрик используется термин concave [20].

Наиболее просто находится наилучший путь с ограничениями, относящимися к классу некуммулятивных.

В то же время QoS маршрутизация при наличии в качестве ограничений аддитивных и мультипликативных метрик характеризуется высокой вычислительной сложностью.

Итак, сформулируем задачу поиска пути со многими ограничениями (задачу MCP) следующим образом.

Пусть сеть представлена в виде направленного графа $G(V, E)$, состоящего из набора вершин (V) и набора ребер (E). Количество вершин графа G равно $n = |V|$. Количество ребер равно $m = |E|$. Каждое ребро представлено в виде связующего звена между вершинами $e = (u, v)$, каждому ребру назначено q весов, соответствующих QoS-метрикам, таких что $\omega_j(u, v) \geqslant 0$ при $j = 1, 2, ..., q$. Ограничение для каждой QoS-метрики — L_j. Задача MCP: найти путь P от источника s до получателя d при условии удовлетворения всех ограничений.

$$\omega_j(P) \leqslant L_j, \quad j = 1, 2, ..., q. \tag{2.17}$$

Назовем пути, которые удовлетворяют этим условиям, допустимыми путями [1]. Решение MCP-задач заключается в поиске алгоритма, который позволит определить все пути, удовлетворяющие уравнению (2.17). Для этого используются различные методы математического программирования, изучаемые в дисциплине «Исследование операций». К ним обычно относятся:

1. Линейное программирование: состоит в нахождении экстремального значения линейной функции многих переменных при наличии линейных ограничений, связывающих эти переменные.

2. Нелинейное программирование: целевая функция и ограничения могут быть нелинейными функциями.

3. Особым случаем в задачах линейного и нелинейного программирования является случай, когда на оптимальные решения накладывается условие целочисленности. Такие задачи относятся к целочисленному программированию.

4. Динамическое программирование: для отыскания оптимального решения планируемая операция разбивается на ряд шагов (этапов) и планирование осуществляется последовательно от этапа к этапу. Однако выбор метода решения на каждом этапе производится с учетом интересов операции в целом.

5. Стохастическое линейное программирование. Бывает много практических ситуаций, когда коэффициенты целевой функции, коэффициенты в матрице коэффициентов, коэффициенты ограничений являются случайными величинами. В этом случае сама целевая функция становится случайной величиной, и ограничения типа неравенств могут выполняться лишь с некоторой вероятностью. Приходится менять постановку самих задач с учетом этих эффектов и разрабатывать совершенно новые методы их решения. Соответствующий раздел получил название стохастического программирования.

6. Геометрическое программирование. Под задачами геометрического программирования понимают задачи наиболее плотного расположения некоторых объектов в заданной двумерной или трехмерной области. Имеющиеся здесь алгоритмы в основном ориентированы на

сокращение перебора вариантов с поиском локальных минимумов.

7. Задачами теории массового обслуживания является анализ и исследование явлений, возникающих в системах обслуживания. Одна из основных задач теории заключается в определении таких характеристик системы, которые обеспечивают заданное качество функционирования, например, минимум времени ожидания, минимум средней длины очереди.

8. Марковские процессы принятия решений. Здесь процесс принятия решений представляется конечным числом состояний. Переходные вероятности между состояниями описывают марковскую цепь.

9. Игровые методы обоснования решений. Эти методы используются в условиях, «когда некоторые параметры, от которых зависит успех операции, неизвестны и нет никаких данных, позволяющих судить о том, какие их значения более, а какие — менее вероятны» [21].

Для первоначального ознакомления с задачами, методологическими принципами и рабочими приемами науки «Исследование операций» можно рекомендовать учебное пособие [21].

Во многих случаях решение задач выбора наилучших путей в задачах с множеством ограничений требует большого объема вычислений, который растет с увеличением числа узлов и звеньев сети. Поэтому обычно используются эвристические или приближенные алгоритмы [22–26] и др.

2.5. Контрольные вопросы

1. Что понимается под качеством обслуживания (QoS)?
2. Какими показателями оценивается качество обслуживания?
3. Перечислите методы обеспечения качества обслуживания.
4. Механизм поддержки QoS в IP-сетях IntSerf и его отличие от DiffSerf.
5. Как зависит кратность исправляемых ошибок от кодового расстояния?
6. Как зависит кратность обнаруживаемых ошибок от кодового расстояния?
7. Принцип построения систематического кода с заданным кодовым расстоянием.
8. Что дает использование в системах передачи данных обратной связи?
9. Виды обратной связи.
10. Чем отличается резервирование от динамического восстановления?
11. Как определяется время восстановления?
12. Дайте характеристику моделям резервирования.
13. Метрики, используемые в задачах маршрутизации.

2.6. Список литературы

1. *Bock R.D., Bargmann R.E.* Analysis of covariance structures // Psychometrika. — 1966. — 31. – Pp.507–533.
2. *Олифер В., Олифер Н.* Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. — 4-е изд. — СПб.: Питер, 2010. — 944 с.
3. [RFC-2210] *Вроцлавский Дж.* Использование RSVP совместно с интегрированными услугам. — Сентябрь, 1997.
4. [RFC-2475] *Блэйк С., Блэк Д., Карлсон М., Дэвис Э., Ванг З., Вайс В.* Архитектура дифференцированных услуг. — Декабрь, 1998.
5. *Bircan G., Cannington J., Ortynski E.A., Spiride G.* Design strategies for meeting unavailability targets using dedicated protection in DWDM networks // IEEE/OSA J. Lightwave Technology. — May 2007. — Vol. 25, № 5. — P.1120–1129.
6. *Шувалов В.П., Тимченко С.В.* Методы резервирования и восстановления в телекоммуникационных сетях // Межвузовский тематический сборник научных трудов. — Омск, 2009. — С.40–44.
7. *Calle E., Marzo J.L., Urra A.* Protection Performance Components in MPLS Networks // Elsevier Computer Communications Journal. — July 2004. — Vol. 27, Issue 12. — P.1220–1228.
8. *Сергеева Т.П., Тетёкин Н.Н.* Методы повышения надежности в сетях SDN // T-Comm — Телекоммуникации и транспорт. — 2014. — Т. 8, вып. 6. — С.53–55.
9. ГОСТ Р 53111-2008. Устойчивость функционирования сети связи общего пользования. Требования и методы проверки. — Стандартинформ, 2008.
10. *Калимулина Э.Ю.* Моделирование и анализ надёжности корпоративной сети // Стандарты и качество. 2008. — № 8. — С.96–112.
11. *Ромашкова О.Н., Иванов П.А., Васюк Д.С.* Анализ отказоустойчивости плоскости управления. Спецификация Generalized Multi-Protocol Label Switching ИКСЗТ, № 5. — 2010.
12. *Dieu-Linh Truong, Brigitte Jaumard*: Recent Progress in Dynamic Routing for Shared Protection in Multi-domain Networks // IEEE Communications Magazine. — June 2008. — Vol. 46, № 6. — P.112–119.
13. *Gao D., Zhang H.* Routing pre-configuration for fast and scalable path restoration in DWDM Networks // Photonic Network Commun. — Dec. 2006. — Vol. 12, № 3. — P.321–327.
14. *Iraschko R., Grover W.* A highly efficient path restoration protocol for management of optical network transport integrity // IEEE J.

Sel. Areas Commun. — May 2000. — Vol. 18, № 5. — P.779–794.

15. *Das Ananya, Charles Martel, Mukherjee Biswanath, and Rai Smita*. New Approach to Reliable Multipath Provisioning // J. Opt. Commun. Netw. — January 2011. — Vol. 3, № 1.
16. *Shooman M.* Reliability of computer systems and Networks. Fault Tolerance, Analysis, and Design. — Jonn Wiley & Sons, Inc. New York, 2002. — 546 p.
17. *Grower W.* Mesh-Based Survivable Networks. Options and strategies for Optical, MPLS, SONET, and ATM Networks / Upper Saddie River, NJ: Prentice Hall PTR, 2004.
18. *Cachin C.* Security and Fault-tolerance in Distributed Systems. Course at ETH Zurich, Department of Computer Science, Spring Semester. — 2013.
19. *Avizienis A. Laprie J.-C., Randell B.* Fundamental concepts of dependability // TR, LAAS-New Castl Univercity-UCLA. — 2001.
20. *Avizienis A. Laprie J.-C., Randell B.* Fundamental concepts of dependability. — UCLA CSD Report #010028, 2000.
21. *Вентцель Е.С.* Исследование операций: задачи, принципы, методология: учебное пособие. — 5 изд., стер. — М.: КНОРУС, 2010. — 192 с.
22. *Попков В.К.* Математические модели связности: монография. — Новосибирск: Изд-во ИВМиМГ СО РАН. — 2006. — 490 с.
23. *Додонов А.Г., Ландэ Д.В.* Живучесть информационных систем. — К.: Наукова думка, 2011. — 256 с.
24. *Laprie J.C.* Basic Concepts and Terminology. — Springer-Verlag, 1992.
25. *Шувалов В.П., Егунов М.М.* Гибкое обслуживание запросов на соединение в условиях отказов // Сборник трудов «Надежность функционирования и информационная безопасность телекоммуникационных систем железнодорожного транспорта». — Омск, 2013. — С.22–27.
26. *Laprie J.C., et al.* Introduction to Dependability. [Электронный ресурс]. — Режим доступа: http://www.dis.uniroma1.it/~ciciani/files/I%20-%20Introduction%20to%20Dependability.ppt, свободный.

Глава 3

Локальные сети

3.1. Протоколы LAN

3.1.1. Технология Ethernet (IEEE 802.3)

Ethernet — это самый распространенный на сегодняшний день стандарт локальных сетей. В зависимости от типа физической среды стандарт IEEE 802.3 имеет различные модификации — 10Base-5, 10Base-2, 10Base-T, 10Base-FL, 10Base-FB.

Передача данных для всех вариантов физического уровня технологии Ethernet осуществляется со скоростью 10 Мбит/с, при этом используется манчестерский код.

Все виды стандартов Ethernet (в том числе Fast Ethernet и Gigabit Ethernet) используют один и тот же метод разделения среды передачи данных — множественный метод доступа с контролем несущей и обнаружением коллизий [1].

Общая характеристика стандарта. К общим характеристикам архитектуры сетей стандарта IEEE 802.3 можно отнести:

- информационный блок — кадр;
- размер кадра — до 1518 байт (без учета преамбулы (8 байт) и завершителя кадра (1 байт));
- обмен кадрами — широковещательный с проверкой адресата;
- среда передачи — коаксиальный кабель (тонкий, толстый), витая пара (3, 4, 5-й категории), оптоволоконный кабель;
- доступ к среде передачи — множественный доступ с контролем несущей и обнаружением коллизий (CSMA/CD);
- скорость передачи данных — 10, 100, 1000 Мбит/с;
- физическая топология — «шина», «звезда»;
- логическая топология — «шина»;
- размеры сетей — от нескольких метров до нескольких километров (при использовании повторителей).

В зависимости от среды передачи данных IEEE 802.3 определяет несколько различных стандартов физических подключений локальных сетей, каждый из которых имеет наименование, в котором отражены такие его важнейшие характеристики:

- 10Base5 — толстый коаксиальный кабель;
- 10Base2 — тонкий коаксиальный кабель;
- 10Base-T — неэкранированная витая пара категории 3;
- 10Base-F — волоконно-оптический кабель.

Информация по сети Ethernet передается в виде кадров (рис. 3.1), каждый из которых состоит из 7 частей [2].

Преамбула состоит из 56 бит, предназначенных для синхронизации приемного тракта. Начальный ограничитель (НО) кадра кодируется как 10101011. Он обозначает начало информационной части кадра.

ПРЕАМБУЛА	НО	АП	АО	Длина/тип	ДАННЫЕ	FCS

Рис. 3.1. Формат кадра Ethernet

АП — адрес получателя. Поле длиной 6 байт содержит адрес узла ЛВС, которому предназначено сообщение. Старший (самый левый) бит в первом байте имеет специальное назначение: если он равен нулю, то адрес назначения является физическим адресом и уникален в ЛВС. В соответствии со схемой присвоения имен, принятой фирмой Xerox, первые три байта задают адрес группы, а следующие три байта задают локальный адрес в группе. Если старший бит в первом байте равен 1, то кадр является широковещательным, и тогда остальные байты в этом поле могут адресовать кадр какой-нибудь конкретной группе или всем рабочим станциям в ЛВС. В этом случае все последние биты в этом байте равны 1.

АО — адрес отправителя. Это поле тоже имеет длину 6 байт и идентифицирует узел, отправивший кадр. Старший бит первого байта в этом поле всегда равен 0.

Поле «длина/тип» может определять длину или тип кадра в зависимости от используемого кадра Ethernet. Если поле задает длину, она указывается в двух байтах и определяет длину поля данных в кадре. Если поле задает тип, то содержимое поля указывает на тип протокола верхнего уровня, которому принадлежит данный кадр.

Поле «Данные» может иметь длину от 46 до 1500 байт и содержит данные, составляющие сообщение. Но если длина поля меньше 46 байт, то используется следующее поле — поле заполнения, дополняющее кадр до минимально допустимого значения в 46 байт.

FCS (Frame Control Sequence) — контрольная сумма кадра длиной 4 байта; она содержит остаток, полученный в результате деления, принимаемой информационной последовательности на образующий полином (CRC-32) и служит для выявления ошибок в поле данных [1].

Данный код позволяет обнаружить 99,999999977% всех ошибок в сообщениях длиной до 64 байт. Таким образом, вероятность того, что принимающая станция воспримет испорченный кадр как целый, практически равна нулю.

3.1.2. Технология Token Ring (IEEE 802.5)

Сети Token Ring так же, как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит

из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передается с помощью кадра специального формата, называемого *маркером* или *токеном* (token).

Сети Token Ring работают с двумя битовыми скоростями — 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается. Сети Token Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с.

Технология Token Ring является более сложной технологией, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры — посланный кадр всегда возвращается в станцию-отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например, может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций является активным монитором. Активный монитор (или ведущая станция) выбирается во время инициализации кольцевой архитектуры как станция с максимальным значением MAC-адреса. Если активный монитор выходит из строя, процедура инициализации кольцевой архитектуры повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора [1].

Характеристика стандарта. Общая характеристика архитектуры сетей стандарта IEEE 802.5:

- информационный блок — кадр, размером до 4522 байт (данных не более 4502 байт);
- обмен кадрами — широковещательный с проверкой адресата;
- среда передачи — экранированная и неэкранированная витая пара, а также волоконно-оптический кабель;
- максимальное количество станций в кольце — 260;
- максимальное расстояние между узлами — 100 метров;

- доступ к среде передачи — маркерный метод или метод передачи права;
- скорость передачи данных — 4 или 16 Мбит/с;
- физическая топология — «звезда»;
- логическая топология — «кольцо»;
- сеть может строиться на основе нескольких колец, разделяемых мостами:
- размеры сетей — максимально 4 км (при использовании повторителей между многопортовыми модулями доступа).

3.1.3. Технология FDDI

Технология FDDI (Fiber Distributed Data Interface — оптоволоконный интерфейс распределенных данных) — это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи [1, 2].

Характеристика стандарта. Общая характеристика архитектуры FDDI следующая:

- информационный блок — кадр;
- размер кадра — до 4500 байт;
- среда передачи — волоконно-оптический кабель (может быть экранированная витая пара);
- максимальная длина сети без мостов до 200 км;
- доступ к среде передачи — маркерный метод или метод логического кольца;
- скорость передачи данных — 100 Мбит/с;
- физическая топология — «звезда», «двойное кольцо»;
- логическая топология — «кольцо»;
- поддерживает до 500 станций (1000 соединений);
- размеры сетей — до 100 км при расстоянии между узлами до 2 км.

Отличие FDDI от IEEE 802.5 заключается в топологии. В FDDI реализована физическая топология «двойное кольцо», при этом допускается физическое подключение отдельных узлов по топологии «звезда». В связи с этим в стандарте определены два типа узлов сети FDDI — однократно подключенные и двукратно подключенные.

Другим принципиальным отличием FDDI является способ управления маркером. Ниже перечислены основные особенности технологии FDDI:

- по сети циркулирует несколько маркеров одновременно (от 2 до 8);
- станция-источник не ждет подтверждения на отправленное со-

общение о его приеме, а посылает маркер сразу после отправления кадра сообщения;
- достигается более высокое быстродействие сети при использовании сокращенного маркера;
- протокол не позволяет сообщениям с низким приоритетом «засорять» сеть в час пик. В этом протоколе предусмотрена как синхронная, так и асинхронная передача;
- аппаратная избыточность, обусловленная наличием двух колец с линиями связи (основное и дополнительное). Информационные потоки (кадры) в кольцах ориентированы в противоположных направлениях. В случае отказа одного кольца сеть автоматически реконфигурируется, и данные начинают передаваться по второму кольцу в другом направлении, как показано на рис. 3.2 [1–4];
- используемые концентраторы, которые предотвращают обрыв цепи в случае неисправности кабеля, могут подключать от четырех до шестнадцати станций [1];
- в отличие от IEEE 802.5, маркер «захватывается» на определенный интервал времени, в течение которого узел может формировать и передавать кадры в сеть. Маркер освобождается в двух случаях: либо истек временной интервал, либо узел закончил передачу данных. Эти два решения и использование в качестве среды передачи данных волоконно-оптического кабеля позволили достичь скорости передачи данных 100 Мбит/с.

Схемы реконфигурации сети при различных неисправностях показаны на рис. 3.3.

Управление доступом к среде MAC осуществляется на основе метода кольцевых слотов.

Более подробную информацию о технологиях Token Ring и FDDI можно найти на сайте olifer.co.uk.

3.1.4. Fast Ethernet (IEEE 802.3u)

Fast Ethernet работает на скорости в 100 Мбит/с. Рассмотрим структуру Fast Ethernet (рис. 3.4) [1–5].

В спецификации IEEE 802.3u функции канального уровня разбиты на два подуровня: управления логической передачей данных (LLC) и доступа к среде (MAC).

Подуровень логической передачи данных (Logical Link Control, LLC), функции которого определены стандартом IEEE 802.2, фактически обеспечивает взаимосвязь с протоколами более высокого уровня, (например, с IP), предоставляя различные коммуникационные услуги:

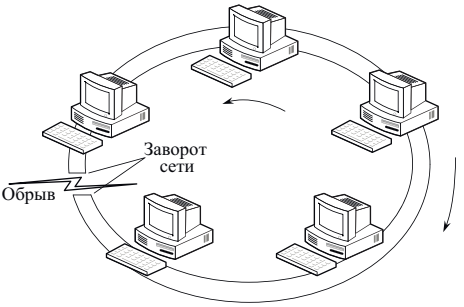


Рис. 3.2. Топология «двойное кольцо» и обход поврежденного участка сети

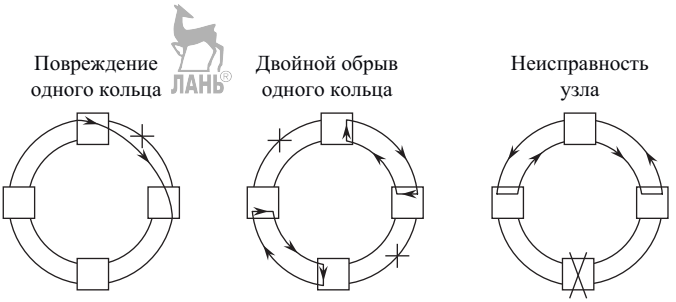


Рис. 3.3. Обход мест повреждения сети

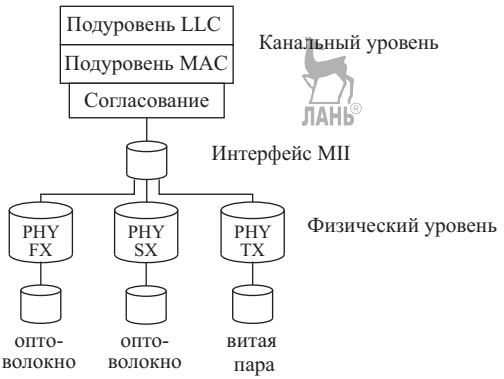


Рис. 3.4. Структура Fast Ethernet

DSAP	SSAP	Поле управления

Рис. 3.5. Заголовок Fast Ethernet

– сервис без установления соединения и подтверждений приема и не гарантирует доставку данных;
– сервис с установлением соединения. Гарантирует правильную доставку данных, за счет установления соединения с системой-приемником до начала передачи данных и использования механизмов контроля ошибок и управления потоком данных;
– сервис без установления соединения с подтверждениями приема. Использует сообщения подтверждения приема для обеспечения гарантированной доставки, но не устанавливает соединения до передачи данных.

Заголовок LLC состоит из трех полей (рис. 3.5).

DSAP (Destination Service Access Point) — составляет 1 байт и является точкой доступа к сервису системы-получателя; указывает, в каком месте буферной памяти системы-получателя следует разместить данные пакета. Однобайтные адреса точек входа определенны стандартом IEEE-802.2 (например, IP — 6, NetBios — 0F).

SSAP (Source Service Access Point) — составляет 1 байт и является точкой доступа к сервису системы-источника; выполняет такие же функции для источника данных, размещенных в пакете, на передающей системе.

Поле управления (Control) состоит из 1 или 2 байт, указывает на тип сервиса, необходимого для данных протокольного блока данных и функций пакета. Определяется типом кадра:

0 — информационный (Information I-frame, 2 байта);

10 — управляющий (Supervisory S-frame, 2 байта);

11 — ненумерованный (Unnumbered, U-frame, 1 байт).

Подуровень управления доступом к среде (Medium Access Control, MAC) имеет три назначения:

– организует правила доступа к среде передачи;
– пересылает кадры физическому уровню (Physical layer, PHY) для преобразования в биты и передачи в среде;
– получает кадры от уровня PHY и передает обрабатывающему их программному обеспечению (протоколам и приложениям).

Подуровень согласования необходим для преобразования интерфейса AUI (Attachment Unit Interface) — интерфейса модуля присоединения в интерфейс MII (Media Independet Interface) — интерфейс независящий от среды передачи.

Устройство физического уровня (Physical layer, PHY). Поскольку Fast Ethernet может использовать различный тип кабеля, то для каждой среды требуется уникальное предварительное преобразование сигнала. Преобразование также требуется для эффективной передачи данных: необходимо сделать передаваемый код устойчивым

к помехам, возможным потерям, либо искажениям отдельных его элементов, обеспечить синхронизацию тактовых генераторов на передающей или приемной стороне.

Стандарт Fast Ethernet определяет тип среды передачи сигналов Ethernet со скоростью 100 Мбит/с:

- 100BASE-T — общий термин для обозначения стандартов, использующих в качестве среды передачи данных витую пару. Длина сегмента до 100 метров. Включает в себя стандарты 100BASE-TX, 100BASE-T4 и 100BASE-T2.
- 100BASE-TX, IEEE 802.3u — развитие стандарта 10BASE-T для использования в сетях топологии «звезда». Задействована витая пара категории 5, фактически используются только две неэкранированные пары проводников, поддерживается дуплексная передача данных, расстояние до 100 м.
- 100BASE-SX — стандарт, использующий одномодовое волокно. Максимальная длина сегмента 400 метров в полудуплексе (для гарантированного обнаружения коллизий) или 2 километра в полном дуплексе.
- 100BASE-FX — стандарт, использующий многомодовое волокно. Максимальная длина ограничена только величиной затухания в оптическом кабеле и мощностью передатчиков, по разным материалам от 2 до 10 километров.
- 100BASE-FX WDM — стандарт, использующий одномодовое волокно. Максимальная длина ограничена только величиной затухания в волоконно-оптическом кабеле и мощностью передатчиков. Интерфейсы бывают двух видов, отличаются длиной волны передатчика и маркируются либо цифрами (длина волны) либо одной латинской буквой A(1310) или B(1550). В паре могут работать только парные интерфейсы: с одной стороны передатчик на 1310 нм, а с другой — на 1550 нм.

Взаимодействие узлов сети. Узлы взаимодействуют друг с другом путем обмена кадрами (frames). В Fast Ethernet кадр является базовой единицей обмена по сети — любая информация, передаваемая между узлами, помещается в поле данных одного или нескольких кадров. Пересылка кадров от одного узла к другому возможна лишь при наличии способа однозначной идентификации всех узлов сети. Поэтому каждый узел в ЛВС имеет адрес, который называется его MAC-адресом. Этот адрес уникален: никакие два узла локальной сети не могут иметь один и тот же MAC-адрес. Более того, ни в одной из технологий ЛВС никакие два узла в мире не могут иметь одинаковый MAC-адрес. Любой кадр содержит, по крайней мере, три основные порции информации: адрес получателя, адрес отправителя и данные.

Некоторые кадры имеют и другие поля, но обязательными являются лишь три перечисленные. На рис. 3.6 отражена структура кадра Fast Ethernet [5].

Минимальный объем кадра составляет 64 байт, или 512 битов. Максимальный объем кадра равен 1518 байт, или 12 144 бита. Кадр содержит поля:

- адрес получателя — указывается адрес узла, получающего данные;
- адрес отправителя — указывается адрес узла, пославшего данные;
- длина/тип (Length/Type, L/T) — содержится информация о типе передаваемых данных;
- контрольная сумма кадра — предназначена для проверки корректности полученного принимающим узлом кадра.

Каждый узел в сети Fast Ethernet имеет уникальный номер, который называется MAC-адресом или адресом узла. Этот номер состоит из 48 битов (6 байтов), присваивается сетевому интерфейсу во время изготовления устройства и программируется в процессе инициализации. Чтобы облегчить процесс управления сетевыми интерфейсами, IEEE было предложено разделить 48-битовое поле адреса на четыре части. Первые два бита адреса (биты 0 и 1) являются флажками типа адреса. Значение флажков определяет способ интерпретации адресной части (биты 2–47).

В поле данных содержится информация, которую один узел пересылает другому. В отличие от других полей, хранящих весьма специфические сведения, поле данных может содержать почти любую информацию, лишь бы ее объем составлял не менее 46 и не более 1500 байтов. Как форматируется и интерпретируется содержимое поля данных, определяют протоколы.

Если необходимо переслать данные длиной менее 46 байт, то используется следующее поле, чтобы дополнить кадр до минимально допустимого значения 46 байт.

Если кадр имеет тип 802.3, то в поле L/T указывается значение объема действительных данных. Например, если пересылается 12-байтовое сообщение, то поле L/T хранит значение 12, а в поле данных находятся и 34 добавочных незначащих байта. Добавление

Адрес получателя	Адрес отправителя	Длина / Тип (L\T)	Данные	Контрольная сумма кадра
6 байтов	6 байтов	2 байта	от 46 до 1500 байтов	4 байта

Рис. 3.6. Структура кадра Fast Ethernet

незначащих байтов инициирует уровень LLC Fast Ethernet и обычно реализуется аппаратно.

Средства уровня MAC не задают содержимое поля L/T — это делает программное обеспечение. Установка значения этого поля почти всегда производится драйвером сетевого интерфейса [7].

3.1.5. Технология 100VG-AnyLAN

Путем объединения стандартов Ethernet и Token Ring создан стандарт 100VG-AnyLan, т.е. технология для любой сети (стандарт IEEE 802.12). Данная технология использует метод доступа с запросом приоритета (Demand priority) синхронных приложений. В отличие от метода доступа CSMA/CD, кадры передаются конкретно станции назначения, что обеспечивает более справедливое распределение пропускной способности сети. В сети имеется корневой концентратор (рис. 3.7) [1], поэтому в такой сети нет коллизий, т.к. применяется распределенный между станциями сети алгоритм доступа. Данные в такой сети передаются одновременно по четырем парам кабеля UTP категории 3. По каждой паре данные передаются со скоростью 25 Мбит/с, что в сумме дает 100 Мбит/с.

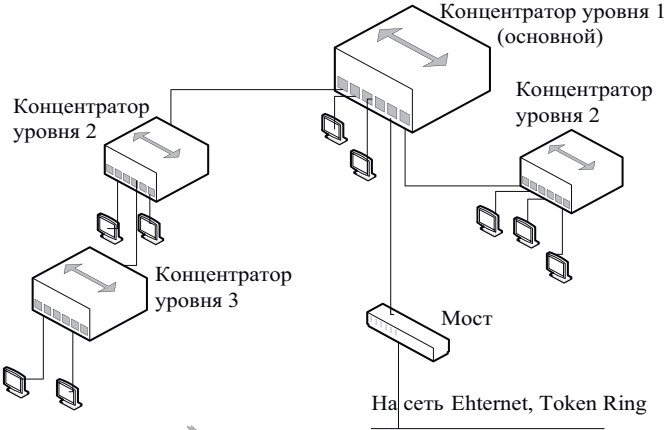


Рис. 3.7. Сеть 100VG-AnyLAN

Допускаются три уровня иерархии. Каждый концентратор и сетевой адаптер 100VG-AnyLAN должен быть настроен либо на работу с кадрами Ethernet, либо с кадрами Token Ring, причем одновременно циркуляция обоих типов кадров не допускается.

Концентратор циклически выполняет опрос портов. Станция, желающая передать пакет, посылает специальный низкочастотный сигнал концентратору, запрашивая передачу кадра и указывая его приоритет. В сети 100VG-AnyLAN используются два уровня приорите-

тов — низкий и высокий. Низкий уровень приоритета соответствует обычным данным (файловая служба, служба печати и т.п.), а высокий приоритет соответствует данным, чувствительным к временным задержкам (например, мультимедиа). Приоритеты запросов имеют статическую и динамическую составляющие, то есть станция с низким уровнем приоритета, долго не имеющая доступа к сети, получает высокий приоритет [1].

Если сеть свободна, то концентратор разрешает передачу пакета. После анализа MAC-адреса получателя в принятом пакете концентратор автоматически отправляет пакет станции назначения. Если сеть занята, концентратор ставит полученный запрос в очередь, которая обрабатывается в соответствии с порядком поступления запросов и с учетом приоритетов. Если к порту подключен другой концентратор, то опрос приостанавливается до завершения опроса концентратором нижнего уровня. Станции, подключенные к концентраторам различного уровня иерархии, не имеют преимуществ по доступу к разделяемой среде, так как решение о предоставлении доступа принимается после проведения опроса всеми концентраторами опроса всех своих портов [1].

В технологии 100VG-AnyLAN концентратор при установлении физического соединения выясняет MAC-адрес станции и запоминает его в таблице MAC-адресов, аналогичной таблице моста/коммутатора. Отличие концентратора 100VG-AnyLAN от моста/коммутатора в том, что у него нет внутреннего буфера для хранения кадров. Поэтому он принимает от станций сети только один кадр, отправляет его на порт назначения, и пока этот кадр не будет полностью принят станцией назначения, новые кадры концентратор не принимает, так что эффект разделяемой среды сохраняется. Улучшается только безопасность сети — кадры не попадают на чужие порты и их труднее перехватить [1].

Сегодня стандарт 100VG-AnyLAN представляет только историческую ценность и упоминается только ради полного перечисления всех родственников семейства стандартов Ethernet.

3.1.6. Высокоскоростная технология Gigabit Ethernet

Чем быстрее растут вычислительные мощности современных персональных компьютеров, тем больше становится среднестатистический объем обрабатываемых с их помощью файлов. Соответственно, возникает потребность в пропорциональном увеличении пропускной способности линий связи. В итоге это заметно ускорило процесс эволюции сетевых технологий: не успел окончательно прижиться стандарт 100Base-T, как ему на смену подоспел новый класс локальных сетей, позволяющих передавать информацию со скоростью до гигаби-

та в секунду. Эти сети получили обозначение 1000Base-T и альтернативное название Gigabit Ethernet [9, 15, 17–22].

Технология Gigabit Ethernet описывается двумя стандартами: IEEE 802.3z и IEEE 802.3ab.

Высокоскоростные сети Gigabit Ethernet определены стандартом IEEE 802.3z. В данной технологии:

- сохраняются все форматы кадров Ethernet;
- используется полудуплексная версия протокола, поддерживающая классический метод множественного доступа с контролем несущей и обнаружением коллизий (CSMA/CD);
- применяются все основные виды кабелей, используемых в классической технологии Ethernet, в том числе волоконно-оптический кабель, витая пара категории 5, экранированная витая пара;
- обеспечивается производительность 1 Гбит/с;
- обеспечивается поддержка на физическом уровне набора спецификаций, обеспечивающих связь при длине кабеля не менее 500 м (многомодовый световод), 25 м (медный кабель, предпочтительно 100 м), 3 км (одномодовый световод);
- обеспечивает поддержку топологий типа «звезда»;
- стандарт IEEE 802.3z включает в себя спецификацию на технологию передачи по сети 1000BASE-CX, которая и устанавливает длину экранированного медного кабеля до 25 м;
- технология Gigabit Ethernet заимствует ряд технологических решений из стандарта ANSI волоконно-оптического канала Fiber Channel, используя, по существу, тот же набор кодов;
- технология Gigabit Ethernet с расширением полосы пропускания IEEE 802.3z поддерживает новую разновидность продуктов, способных маршрутизировать трафик различных сетей с гигабитной скоростью передачи данных, т.е. в 100 раз быстрее традиционных программных маршрутизаторов;
- минимальный размер кадра увеличен (без учета преамбулы) с 64 до 512 байт или до 4096 бит;
- для сокращения накладных расходов при использовании слишком длинных кадров для передачи коротких квитанций разработчики стандарта разрешили конечным узлам передавать несколько кадров подряд без передачи среды другим станциям. Такой режим получил название Burst Mode — монопольный пакетный режим. 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). Предел 8192 байт называется Burst Length. Если станция начала передавать кадр и предел Burst Length был достигнут в середине кадра, то кадр разрешается передать до конца.

Увеличение «совмещенного» кадра до 8192 байт несколько задерживает доступ к разделяемой среде других станций, но при скорости 1000 Мбит/с эта задержка не столь существенна [8, 9].

Характеристики производительности Gigabit Ethernet зависят от того, использует ли коммутатор режим передачи кадров с расширением или же передает их в режиме пульсаций. В режиме пульсаций на периоде пульсации характеристики отличаются в 10 раз от характеристик Fast Etherhet:

- максимальная скорость в кадрах в секунду (для кадров минимальной длины с полем данных 46 байт) составляет 148 800;
- полезная пропускная способность для кадров минимальной длины равна 548 Мбит/с;
- полезная пропускная способность для кадров максимальной длины (поле данных 1500 байт) равна 976 Мбит/с [1, 9].

Стандарт IEEE 802.3z имеет несколько версий: 1000BaseSX, 1000BaseLX, 1000BaseCX.

Версия 1000BaseSX определяет работу по многомодовому оптоволокну на длине волны 850 нм. Максимальная длина сегмента при работе в полудуплексном режиме составляет 100 м. В дуплексном режиме максимальная длина кабеля зависит от его полосы пропускания и может достигать 800 м.

Версия 1000BaseLX определяет работу по многомодовому или одномодовому оптоволокну на длине волны 1310 нм. Максимальная длина сегмента для одномодового волокна достигает 5 км, а для многомодового 550 м.

Версия 1000BaseCX использует в качестве среды передачи твинаксиальный (twinaxial) кабель, который представляет собой два коаксиальных кабеля (волновое сопротивление 75 Ом) в общей оплетке. По такому кабелю можно организовать только полудуплексный режим. Максимальная длина сегмента составляет 25 м, потому такой кабель больше всего применяется для связи оборудования в пределах одной комнаты.

В июне 2002 г. утвержден стандарт 10-гигабитный Ethernet (IEEE 802.3ae) для построения региональных каналов. Он соответствует спецификациям OC-192c/SDH VC-4-46c (WAN) (рис. 3.8).

На рис. 3.8:

– RS (Reconciliation Sublayer) — подуровень согласования —

функция отображения, согласующая сигналы перед интерфейсом XGMII, соединяющим с подуровнем MAC и PCS;
- XGMII (10 Gigabit Media Independent Interface) — 10-гигабитный интерфейс, независимый от среды;
- PCS (Physical Coding Sublayer) — верхний подуровень физического кодирования;
- PMA (Physical Medium Attachment) — подуровень подсоединения к физической среде;
- PMD (Physical Medium Dependent) — подуровень физического уровня, зависящий от среды передачи;
- WIS (WAN Interface Sublayer) — подуровень интерфейса сети.
- MDI (Medium Dependent Interface) — интерфейс, зависящий от среды передачи.

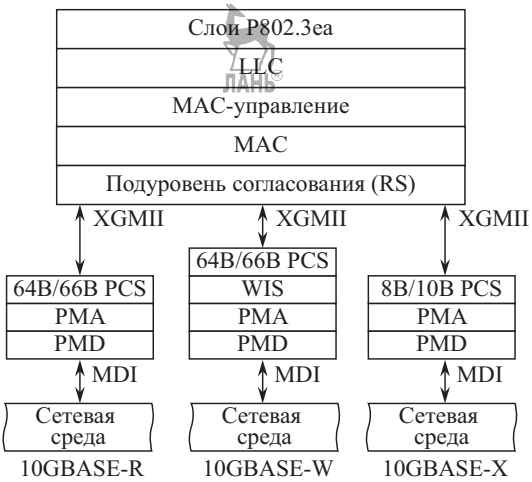


Рис. 3.8. Схема уровней для 10Gbase Ethernet

Спецификация 10GBase-X описывает семейство версий 10GE, использующих четырехпотоковую передачу (в формате 4×8 бит) с кодированием каждого потока кодом 8B/10B. Эта спецификация поддерживается практически всеми уровнями и интерфейсами: MAC, RS, XGMII, PCS, PMA и PMD и может передаваться по медным парам и волоконно-оптическому кабелю. К этому семейству принадлежит версия 10GBase-LX4 — стандарт 10GE для среды передачи на базе волоконно-оптического кабеля, работающей на четырех длинах волн с шагом 13,4 нм во втором окне прозрачности (1300 нм). На каждой длине волны передается один из четырех потоков данных (Lane). Потоки объединяются мультиплексором WDM на передающей стороне перед подачей в волоконно-оптический кабель демультиплексируются

на приемной стороне [9].

Спецификация 10GBase-R — это семейство версий (10GBase-SR, 10GBase-LR и 10GBase-ER), работающих на волоконно-оптическом кабеле в трех разных окнах прозрачности: 850 нм (S), 1300 нм (L) и 1550 нм (E), соответственно. Эти спецификации используются либо самостоятельно (после кодирования данных на подуровне PCS по схеме 64B/66B), либо они могут превращаться в спецификации 10GBase-W (если потоки данных после PCS передаются в глобальную сеть передачи данных — интерфейсу WIS) [9].

Спецификация 10GBase-W — это семейство из трех версий: 10GBase-SW, 10GBase-LW и 10GBase-EW, также работающих через волоконно-оптический кабель в трех разных окнах прозрачности: 850 нм (S), 1300 нм (L) и 1550 нм (E). В соответствии с этими спецификациями, потоки указанных версий после кодирования в подуровне PCS по схеме 64B/66B подключаются к глобальной сети передачи данных — интерфейсу WIS, чтобы далее инкапсулироваться в кадры, формируемые в технологиях SONET и SDH для транспорта потоков 10GBASE-SW, 10GBASE-LW и10GBASE-EW через физический уровень [9].

Перспективы развития Ethernet. Согласно наблюдениям Группы 802.3ba, требования к полосе пропускания для вычислительных задач и приложений ядра сети растут с разными скоростями, что определяет необходимость двух соответствующих стандартов для следующих поколений Ethernet — 40 Gigabit Ethernet (или 40GbE) и 100 Gigabit Ethernet (или 100GbE).

Основные задачи, которые должен решать стандарт 40/100GE:

- поддерживать только полнодуплексные режимы Ethernet MAC-уровня;
- сохранять формат кадра Ethernet 802.3 MAC-уровня;
- сохранять минимальный и максимальный размеры кадров стандарта IEEE 802.3;
- обеспечивать поддержку коэффициента битовых ошибок на уровне не хуже 10^{-12} на интерфейсе между MAC- и физическим уровнями;
- обеспечивать совместимость с оптическими транспортными сетями (OTN, Optical Transport Network);
- поддерживать скорость 40 и 100 Гбит/с на MAC-уровне.

Стандарт определяет восемь типов интерфейсов физического уровня для различных сред и скоростей передачи (40 и 100 Гбит/с). В качестве среды передачи предусматриваются одномодовое и многомодовое оптическое волокно, медная витая пара и электрическая объединительная шина (backplane). Скорость 100 Гбит/с для работы

в объединительных шинах не предусмотрена. На скорости 100 Гбит/с предусмотрены дальний (LR, Longer Rang — до 10 км) и сверхдальний (ER, Extended Range — до 40 км) режимы работы. Отличаются они только требованиями к бюджету линии — сверхдальний режим фактически означает необходимость применения в ВОЛС оптических усилителей.

Стандарт предоставил возможность перехода от последовательной передачи сигнала к параллельной по нескольким потокам (lanes). Скорость в каждом физическом канале — либо 10, либо 25 Гбит/с. Так, для 100-Гбит/с интерфейсов возможна передача десяти 10-Гбит/с потоков либо по 10 витым медным парам, либо по 10 многомодовым оптическим волокнам в каждом направлении.

Для работы по одномодовому оптическому волокну используются четыре потока по 25 Гбит/с, разделенные по длине волны и передаваемые в одном оптическом волокне. Для 40-Гбит/с интерфейсов используются только 10-Гбит/с потоки, передаваемые по проводникам объединительной шины, по медным парам или по оптическому волокну [9].

Стандарт позволяет инкапсулировать Ethernet-потоки в транспортные потоки сетей OTN соответствующих уровней: 10GE — в оптический транспортный модуль (OTU2, Optical Transport Unit — 10,71 Гбит/с), 40GE — в OTU3 (43,02 Гбит/с), 100GE (103,12 Гбит/с) — в OTU4 (111,81 Гбит/с).

Следующим шагом развития может стать появление технологии 400GE. Все предпосылки к этому есть, так как многопотоковая структура 40/100GE вполне допускает масштабирование. Причем масштабирование возможно как по числу потоков (линий передачи, несущих в одномодовом оптическом волокне), так и по скорости отдельного трансивера в потоке. Например, мультиплексируя 16 потоков по 25 Гбит/с, можно достичь требуемой скорости в 400 Гбит/с, а 40 таких потоков дадут уже 1 Тбит/с.

Таким образом, стандарт IEEE 802.3ab — это новое поколение Ethernet-технологий, открывающий путь к дальнейшему росту скоростей.

3.2. Технические средства, обеспечивающие функционирование высокоскоростных сетей передачи данных

3.2.1. Концентраторы

Сетевой концентратор или *хаб* (hub — центр деятельности) — сетевое устройство для объединения нескольких устройств Ethernet в общий сегмент. Все порты концентратора равноправны. Получив сигнал от одной из подключенных к нему станций, концентратор транс-

лирует его на все свои активные порты. Устройства подключаются при помощи витой пары, коаксиального кабеля или оптоволокна.

Термин концентратор используется вместо термина «повторитель», когда речь идет об устройстве, которое служит центром сети (рис. 3.9).

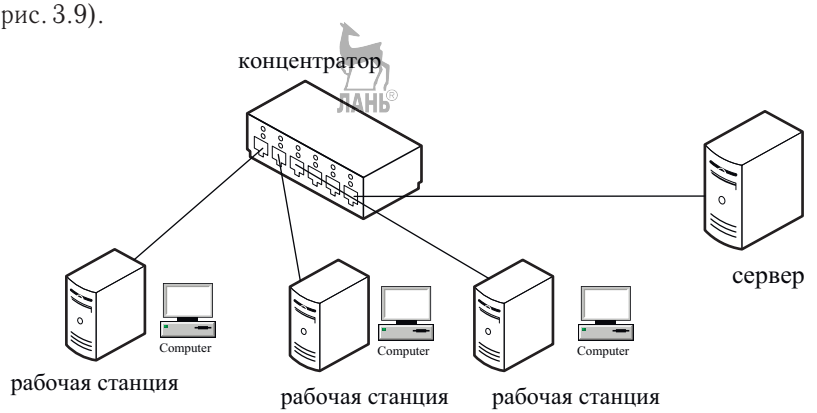


Рис. 3.9. Концентратор объединяет все подключенные к нему устройства

Наиболее важные особенности концентраторов:

- усиливают сигналы;
- распространяют сигналы в сети;
- используются как точки концентрации в сети.

Дополнительные функции концентратора:

1. *Отключение порта*. Концентраторы способны отключать некорректно работающие порты. Эту функцию называют автосегментацией. Отключение происходит:
   - при отсутствии ответа на тест, посылаемый во все порты каждые 16 мс;
   - если интенсивность прохождения через порт кадров с ошибками превышает заданный порог. Виды ошибок: неверная контрольная сумма; неверная длина кадров (больше 1518 байт или меньше 64 байт); неоформленный заголовок кадров; множественные коллизии (если концентратор фиксирует, что источником коллизии является один и тот же порт 60 раз подряд, то порт отключается);
   - затянувшаяся передача (если время прохождения одного кадра через порт превышает время передачи кадра максимальной длины в 3 раза, то порт отключается).

2. *Поддержка резервных связей*. При конфигурировании концентратора администратор определяет, какие порты активные и какие резервные. Если основной порт отключается, то резервный порт становится активным.

Для концентратора FDDI эта функция для многих ошибочных ситуаций является основной.

Если между станциями произошел разрыв кабеля, то передача может произойти через станцию в обратном направлении. Рабочая станция использует ту часть потока, которая исправна и концентратор осуществляет замыкание кольца и передает всю информацию по одному каналу в обход неисправного пути сети [1].

Концентратор можно представить себе в виде устройства, которое содержит множество независимых, но связанных между собой модулей сетевого оборудования. В локальных сетях концентраторы ведут себя как мультипортовые повторители. В таких случаях концентраторы используются, чтобы разделить сетевые носители и обеспечить множественное подключение.

Недостатком использования концентратора является то, что он не может фильтровать сетевой трафик. Фильтрацией называется процесс, в ходе которого в сетевом трафике контролируются определенные характеристики, например, адрес источника, адрес получателя или протокол, и на основании установленных критериев принимается решение — пропускать трафик дальше или игнорировать его.

Концентратор Ethernet может иметь от 8 до 72 портов, большая часть которых предназначена для подключения кабеля на витой паре. По конструктивным особенностям различают концентраторы:

1. *С фиксированным количеством портов*. Общее количество портов от 8 до 24. Один порт специально выделен для подключения одного порта к другому. Имеется разъем для соединения с толстым коаксиальным кабелем, концентратором оптоволоконных сетей.

2. *Модульные концентраторы*. Выполнены в виде отдельных модулей с фиксированным количеством портов, установленных на общие шасси, которые имеют внутреннюю шину для объединения отдельных модулей в единый концентратор. Чаще всего такие концентраторы бывают многосегментными, тогда в пределах одного концентратора работает несколько независимых между собой повторителей. Такие концентраторы снабжаются источником питания и системой терморегуляции, что позволяет заменять модули без отключения питания.

3. *Стековые концентраторы*. Выполняются в виде отдельного корпуса с фиксированным количеством портов. Число объединяемых в стек корпусов бывает от 8 и выше.

4. *Модульно-стековые концентраторы*. Это модульные концентраторы, которые объединяются при помощи специальных связей в стек. Концентраторы такого типа имеют корпуса, рассчитанные на сравнительно количество модулей — примерно от одного до трех.

3.2.2. Мосты

Предназначены для соединения сетевых сегментов, имеющих различные физические среды. Мосты также могут быть использованы для связи сегментов, имеющих различные протоколы низкого уровня (физического и канального). Возможно применение мостов для связи сегментов ЛВС, как с одинаковыми протоколами, так и для связи сегментов осуществляющих соединение с различными протоколами.

Задачи мостов:

- передача пакетов из одной сети в другую и наоборот. В процессе передачи мост регенерирует пакет, что позволяет передавать данные вдоль сети на значительное расстояние;
- просмотр каждого пакета и принятие решения, какой из двух сетей принадлежит тот или иной пакет;
- отслеживание адреса приемника и передатчика информации в процессе передачи какого-либо пакета;
- определение, какой сети принадлежит тот или иной пакет благодаря просмотру информации уровня управления доступом к среде передачи.

Мосты целесообразно применять, когда:

- требуется повысить производительность ЛВС. Это достигается путем деления одной большой сети на две части;
- для осуществления сопряжения аппаратных средств с различными кабельными соединениями.

Мосты представляют собой одноранговые программно-аппаратные комплексы [14]. В локальных сетях 80-х и 90-х годов применялись мосты нескольких типов [1]:

- прозрачные;
- транслирующие;
- инкапсулирующие;
- с маршрутизацией от источника.

Из всех перечисленных выше типов мостов в результате исчезновения всех технологий локальных сетей, кроме Ethernet, интерес представляют только прозрачные мосты. Ниже дается краткая характеристика этого прозрачного моста. Подробное описание прозрачных мостов можно найти в [1].

Прозрачные мосты нужны для объединения сетей с одинаковыми протоколами на канальном и физическом уровне 10Base2 (рис. 3.10). Эти мосты не нагружают работой остальные устройства, им не надо участвовать в выборе маршрута и фильтрации пакета, так как, с точки зрения сетевых устройств, они находятся в одной большой сети с единым сетевым адресом и разными MAC-адресами.

Мост:

- используя протокол канального и физического уровня сегмента А (рис. 3.10), считывает из заголовков пакетов, передаваемых из этого сегмента MAC, адрес назначения;
- игнорирует пакеты, адресованные в сегмент А;
- используя протоколы канального и физического уровня, которые общие в обоих сегментах, мост передает пакеты из сегмента А в сегмент В. Он должен обладать знаниями о месте нахождения сетевых устройств и передает пакеты в соответствии со своей базой данных;
- при получении пакета сравнивает адрес назначения в базе данных, называемой таблицей передач. Если такого адреса в базе данных нет, то он передает пакет по всем направлениям. Если в базе данных адрес значится, то он сравнивает значения направления из базы данных и от пришедшего пакета. Их совпадение означает, что адреса отправителя и получателя расположены в одном сегменте сети. В этом случае пакет транслировать не нужно и мост его игнорирует, когда же оказывается, что адрес отправителя и получателя расположены на разных направлениях, мост отправляет пакет в нужный сегмент сети [14].

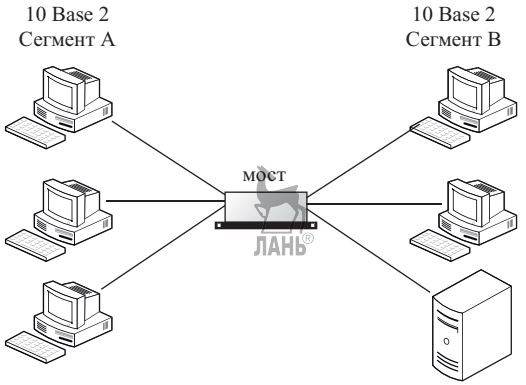


Рис. 3.10. Соединение сетей с использованием прозрачного моста

3.2.3. Коммутаторы

Коммутатор — это устройство, конструктивно выполненное в виде сетевого концентратора и действующего как высокоскоростной много портовый мост.

Коммутаторы обеспечивают широковещательное сегментирование локальных сетей и выделение полосы пропускания к рабочим станциям. Коммутаторы устраняют физические ограничения, возника-

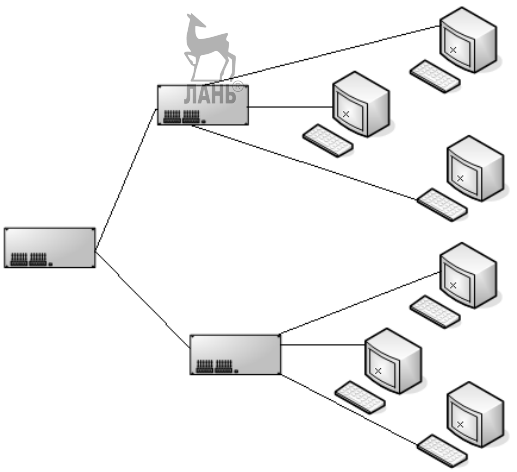


Рис. 3.11. Коммутаторы могут использоваться для группировки пользователей, портов или логических адресов в группы по интересам

ющие вследствие совместного использования концентратора, поскольку они логически группируют пользователей и порты всего предприятия.

Коммутаторы могут быть использованы для создания виртуальных сетей, осуществляющих сегментацию. В традиционных конфигурациях локальных сетей сегментация осуществляется маршрутизаторами [14].

Коммутаторы являются основными компонентами, обеспечивающими обмен данными в виртуальных сетях. Как показано на рис. 3.11, в виртуальной сети они выполняют жизненно важные функции, являясь для устройств конечной станции точкой входа в среду коммутации, а также обеспечивают обмен данными в рамках всего предприятия.

Каждый коммутатор обладает способностью принимать решения о фильтрации и отправке фреймов (frame) на основе метрики виртуальной сети, определяемой сетевыми администраторами, а также способностью передавать эту информацию другим коммутаторам и маршрутизатором сети.

Существует большое разнообразие моделей коммутаторов. Они отличаются как внутренней организацией, так и набором выполняемых дополнительных функций, таких как трансляция протоколов, поддержка алгоритма покрывающего дерева, образование виртуальных логических сетей и ряда других.

По виду технической реализации различают коммутаторы:

– на центральном процессоре общего назначения;

– на основе коммутационной матрицы;
– с общей шиной;
– с разделяемой памятью;
– комбинированные;
– без буферизации.

Коммутатор на центральном процессоре общего назначения. Данный тип коммутатора изображен на рис. 3.12.

Для связи с интерфейсными портами I/O используется внутренняя скоростная шина.

Основным недостатком таких коммутаторов была их низкая скорость. Универсальный процессор никак не мог справиться с большим объемом специализированных операций по пересылке кадров между интерфейсными модулями [16].

Коммутатор на основе коммутационной матрицы. Коммутационная матрица обеспечивает основной и самый быстрый способ взаимодействия процессоров портов (рис. 3.13). Однако реализация матрицы возможна только для определенного числа портов, причем сложность схемы возрастает пропорционально квадрату количества портов коммутатора.

Базовая архитектура на основе коммутационной матрицы $N \times N$

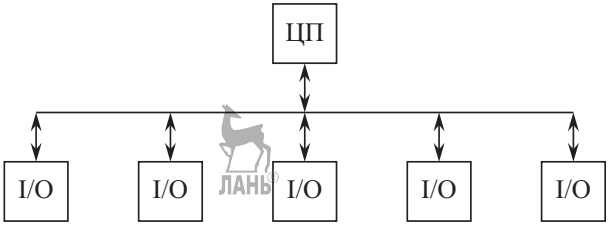


Рис. 3.12. Коммутатор на процессоре общего назначения

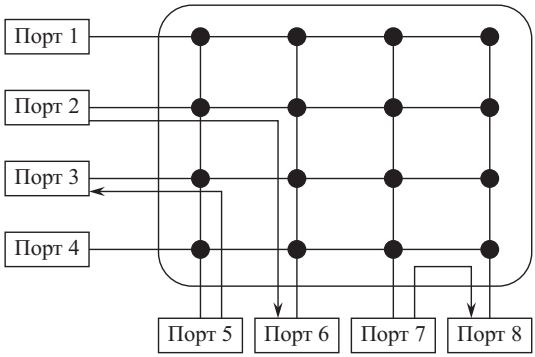


Рис. 3.13. Коммутатор с коммутационной матрицей

обеспечивает соединение N входных портов с N выходными портами в виде матрицы. В местах пересечения проводников, соединяющих входные порты с выходными, находятся коммутирующие устройства, которыми управляет специальный контроллер. В каждый момент времени, анализируя адресную информацию, контроллер сообщает коммутирующим устройствам, какой входной порт должен быть подключен к какому выходному порту. В том случае, если два входящих пакета от разных портов-источников будут переданы на один и тот же выходной порт, он будет заблокирован [16].

Матрица может быть реализована и по-другому, на основании комбинационных схем другого типа, но ее особенностью все равно остается технология коммутации физических каналов.

Недостатки:

1) отсутствие буферизации данных внутри коммутационной матрицы: если составной канал невозможно построить из-за занятости выходного порта или промежуточного коммутационного элемента, то данные должны накапливаться в их источнике, в данном случае — во входном блоке порта, принявшего кадр;
2) сложность наращивания числа коммутируемых портов.

Достоинства — высокая скорость коммутации и регулярная структура, которую удобно реализовывать в интегральных микросхемах.

Коммутаторы с общей шиной. В коммутаторах с общей шиной процессоры портов связывают высокоскоростной шиной (рис. 3.14), используемой в режиме разделения времени. Чтобы шина не блокировала работу коммутатора, ее производительность должна равняться, по крайней мере, сумме производительности всех портов коммутатора.

Кадр должен передаваться по шине небольшими частями, по нескольку байт, чтобы передача кадров между несколькими портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом. Размер такой ячейки данных определяется производителем коммутатора. Входной блок процессора помещает в ячейку, переносимую по шине, тег, в котором указывает номер пор-

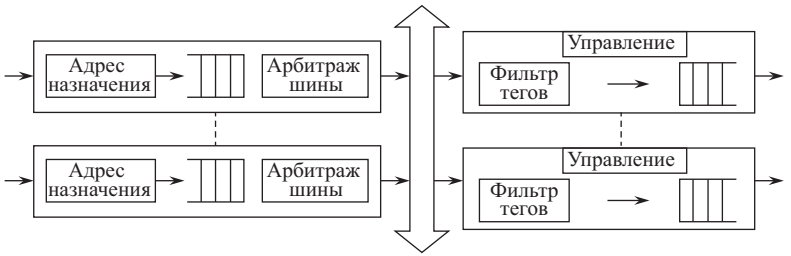


Рис. 3.14. Архитектура коммутатора с общей шиной

та назначения. Каждый выходной блок процессора порта содержит фильтр тегов, который выбирает теги, предназначенные данному порту [16].

Шина так же, как и коммутационная матрица, не может осуществлять промежуточную буферизацию, но так как данные кадра разбиваются на небольшие ячейки, то задержек с начальным ожиданием доступности выходного порта в такой схеме нет — здесь работает принцип коммутации пакетов, а не каналов [16].

Данные коммутаторы используются в основном для трансляции протоколов локальных сетей в протокол ATM, если коммутатор поддерживает эти технологии.

Коммутаторы с разделяемой памятью. Входные блоки процессоров портов соединяются с переключаемым входом разделяемой памяти, а выходные блоки этих же процессоров соединяются с переключаемым выходом этой памяти (рис. 3.15). Переключением входа и выхода разделяемой памяти управляет менеджер очередей выходных портов. В разделяемой памяти менеджер организует несколько очередей данных, по одной для каждого выходного порта. Входные блоки процессоров передают менеджеру портов запросы на запись данных в очередь того порта, который соответствует адресу назначения пакета. Менеджер по очереди подключает вход памяти к одному из входных блоков процессоров и тот переписывает часть данных кадра в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным блокам процессоров портов, и данные из очереди переписываются в выходной буфер процессора [16].

Память должна быть достаточно быстродействующей для поддержания скорости переписи данных между N портами коммутатора. Применение общей буферной памяти, гибко распределяемой менеджером между отдельными портами, снижает требования к размеру буферной памяти процессора порта [16].

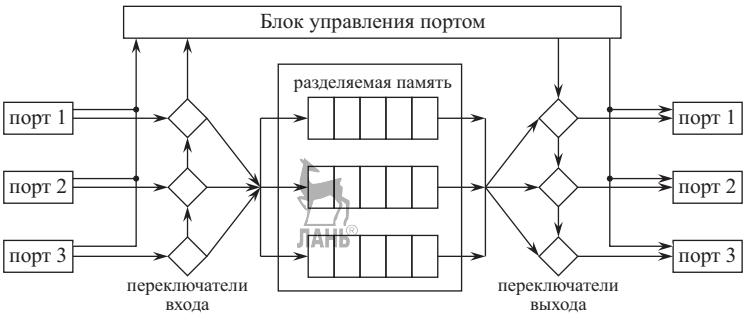


Рис. 3.15. Коммутатор с разделяемой памятью

Комбинированные коммутаторы. У каждой из описанных выше архитектур есть свои преимущества и недостатки, поэтому часто в сложных коммутаторах эти архитектуры применяются в различных комбинациях друг с другом.

В конструктивном исполнении коммутаторы делятся на следующие типы:

1. Автономные коммутаторы с фиксированным количеством портов. Используются в основном для организации небольших рабочих групп.

2. Модульные коммутаторы на основе шасси. Используются на магистральных сетях.

3. Коммутаторы с фиксированным количеством портов, собираемые в стек. Это коммутаторы, которые могут работать автономно, так как выполнены в отдельном корпусе, но имеют специальные интерфейсы, которые позволяют их объединять в общую систему, работающую как единый коммутатор (образуют стек). Обычно такой специальный интерфейс представляет собой высокоскоростную шину. Стековые коммутаторы применяются для создания сетей рабочих групп и отделов.

Существует несколько способов коммутации.

Коммутация с промежуточным хранением. При коммутации с промежуточным хранением (store-and-forward) — коммутатор копирует весь принимаемый кадр в буфер и производит его проверку на наличие ошибок. Если кадр содержит ошибки, то он отбрасывается. Если кадр не содержит ошибок, то коммутатор находит адрес приемника в своей таблице коммутации и определяет исходящий интерфейс. Затем, если не определены никакие фильтры, он передает этот кадр приемнику [16].

Этот способ передачи связан с задержками: чем больше размер кадра, тем больше времени требуется на его прием и проверку на наличие ошибок.

Коммутация без буферизации. Коммутатор локальной сети копирует во внутренние буферы только адрес приемника (первые 6 байт после префикса) и сразу начинает передавать кадр, не дождавшись его полного приема. Этот режим уменьшает задержку, но проверка на ошибки в нем не выполняется. Существует две формы коммутации без буферизации:

- *коммутация с быстрой передачей* (fast-forward switching) — эта форма коммутации предполагает низкую задержку за счет того, что кадр начинает передаваться медленно, как только будет прочитан адрес назначения. Передаваемый кадр может содержать ошибки. В этом случае сетевой адаптер, которому

предназначен этот кадр, отбросит его, что вызовет необходимость повторной передачи этого кадра [16];

– *коммутация с исключением фрагментов* (fragment-free-switching) — коммутатор фильтрует коллизионные кадры перед их передачей. В правильно работающей сети коллизия может произойти во время передачи первых 64 байт. Поэтому все кадры с длиной больше 64 байт считаются правильными. Этот метод коммутации ждет, пока полученный кадр не будет проверен на предмет коллизии, и только после этого начнет его передачу. Такой метод коммутации уменьшает количество пакетов, передаваемых с ошибками.

Режимы работы коммутаторов ЛВС. Коммутаторы ЛВС поддерживают два режима работы:

– *полудуплексный режим* — это режим, при котором только одно устройство может передавать данные в любой момент времени в одном домене коллизии;

– *дуплексный режим* — это режим работы, который обеспечивает одновременную двустороннюю передачу данных между станцией отправителем и станцией получателем на MAC-уровне [16].

Существует разделение коммутаторов по уровням. Коммутатор 2 уровня (Layer 2). Сюда относят все устройства, которые работают на 2 уровне сетевой модели OSI — канальном уровне. К таким устройствам можно отнести все неуправляемые коммутаторы и часть управляемых.

Коммутаторы 2 уровня работают с данными не как с непрерывным потоком информации (коммутаторы 1 уровня), а как с отдельными порциями информации — кадрами. Они умеют анализировать получаемые кадры и работать с MAC-адресами устройств отправителей и получателей кадра. Такие коммутаторы «не понимают» IP-адреса компьютеров, для них все устройства имеют названия в виде MAC-адресов.

Коммутаторы 2 уровня составляют коммутационные таблицы, в которых соотносят MAC-адреса встречающихся сетевых устройств с конкретными портами коммутатора.

Коммутатор 3 уровня (Layer 3). Сюда относятся все устройства, которые работают на 3 уровне сетевой модели OSI — сетевом уровне. Данные коммутаторы управляемые. Коммутаторы 3 уровня целесообразнее отнести уже не к разряду коммутаторов, а к разряду маршрутизаторов, так как эти устройства уже полноценно могут маршрутизировать проходящий трафик между разными сетями. Коммутаторы 3 уровня полностью поддерживают все функции и стандарты коммутаторов 2 уровня. С сетевыми устройствами они могут работать по

IP-адресам. Коммутатор 3 уровня поддерживает установку различных соединений: VPN, PPP и т.д.

Коммутатор 4 уровня (Layer 4). Такой коммутатор работает на транспортном уровне, умеет работать с приложениями. Коммутаторы 4 уровня используют информацию, которая содержится в заголовках пакетов и относится к уровню 3 и 4 стека протоколов, такую как IP-адреса источника и приемника, номер портов TCP/UDP для идентификации принадлежности трафика к различным приложениям. На основании этой информации коммутаторы 4 уровня могут принимать интеллектуальные решения о перенаправлении трафика того или иного сеанса.

Чтобы правильно подобрать коммутатор, необходимо представлять всю топологию будущей сети, рассчитать примерное количество пользователей, выбрать скорость передачи данных для каждого участка сети и уже под конкретную задачу начинать подбирать оборудование.

3.2.4. Протокол STP

Применение протокола STP. Одним из методов повышения отказоустойчивости компьютерной сети является использование протокола STP (Spanning Tree Protocol). В сетях Ethernet коммутаторы поддерживают только древовидные связи, т.е. не содержащие петель. Это означает, что для организации альтернативных каналов требуются особые протоколы и технологии, выходящие за рамки базовых, к которым относится Ethernet [12].

Протокол STP на основе алгоритма STA (Spanning-Tree Algoritm) позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой.

Коммутаторы, поддерживающие протокол STP, автоматически создают древовидную конфигурацию связей без петель в компьютерной сети. Такая конфигурация называется покрывающим деревом (Spanning Tree). Конфигурация покрывающего дерева строится коммутаторами автоматически с использованием обмена служебными пакетами.

Работа протокола STP. Рассмотрим подробно работу протокола STP [12]. Алгоритм STA требует, чтобы каждому коммутатору был присвоен идентификатор.

Идентификатор коммутатора — 8-байтное поле, которое состоит из двух частей: 2-байтного приоритета, назначенного администратором, и 6-байтного MAC-адреса его блока управления. Каждому порту также назначается уникальный идентификатор в пределах коммутатора, как правило, это его MAC-адрес. Каждому порту коммутатора

ставится в соответствие стоимость маршрута, соответствующая затратам на передачу кадра по локальной сети через данный порт.

Процесс вычисления связующего дерева на *первом этапе* начинается с выбора корневого коммутатора (root switch), от которого будет строиться дерево. В качестве корневого выбирается коммутатор с наименьшим значением идентификатора. (Первоначально, по умолчанию, все коммутаторы имеют одинаковое значение приоритета, равное 32 768. В этом случае корневой коммутатор определяется по наименьшему MAC-адресу). Иногда такой выбор может оказаться далеко не рациональным. Для того, чтобы в качестве корневого моста было выбрано определенное устройство (исходя из структуры сети), администратор может повлиять на процесс выбора, присвоив соответствующему коммутатору наименьший идентификатор вручную [12].

Второй этап работы STP — выбор корневого порта (root port) для каждого из остальных коммутаторов сети.

Корневой порт коммутатора — это порт, который имеет по сети кратчайшее расстояние до корневого коммутатора.

Третий этап работы STP — определение назначенных портов. Каждый сегмент в коммутируемой сети имеет один назначенный порт (designated port). Этот порт функционирует как единственный порт коммутатора, т.е. принимает пакеты от сегмента и передает их в направлении корневого коммутатора через корневой порт данного коммутатора. Коммутатор, содержащий назначенный порт для данного сегмента, называется назначенным коммутатором (designated bridge) этого сегмента. Назначенный порт сегмента имеет наименьшее расстояние до корневого коммутатора среди всех портов, подключенных к данному сегменту. Назначенный порт у сегмента может быть только один. У корневого коммутатора все порты являются назначенными, а их расстояние до корня полагается равным нулю. Корневого порта у корневого коммутатора нет.

При построении покрывающего дерева важную роль играет понятие расстояния. По этому критерию выбирается единственный порт, соединяющий каждый коммутатор с корневым коммутатором, и единственный порт, соединяющий каждый сегмент сети с корневым коммутатором. Все остальные порты переводятся в резервное состояние, то есть такое, при котором они не передают обычные кадры данных. При таком выборе активных портов в сети исключаются петли и оставшиеся связи образуют покрывающее дерево [12].

В качестве расстояния в STA (Spanning-Tree Algoritm) используется метрика стоимость пути (Path Cost); она определяется как суммарное условное время на передачу данных от порта данного коммутатора до порта корневого коммутатора. Условное время сегмента

рассчитывается как время передачи одного бита информации через канал с определенной полосой пропускания. В табл. 3.1 приводятся типичные стоимости пути в соответствии со стандартом IEEE 802.1d:

Таблица 3.1. Определение стоимости пути

Параметр	Скорость канала	Рекомендованное значение
Стоимость пути	10 Мбит/сек	100
Стоимость пути	100 Мбит/сек	19
Стоимость пути	1 Гбит/сек	4
Стоимость пути	10 Гбит/сек	2

Пример работы протокола STP рассмотрены на рисунках рис. 3.16 и 3.17. На рис. 3.16 представлена общая схема, на которой обозначены номера коммутаторов SW1, SW2, SW3, номера портов 1, 2, 3 и стоимость пути в кружочке, а также приоритетность портов коммутатора 10, 20, 30.

Для выполнения задания поэтапно:

1. Определим корневой коммутатор (рис. 3.17). В качестве корневого коммутатора выбираем SW1, так как данный коммутатор с наименьшей приоритетностью;

2. Определим порты SW2 и SW3, которые имеют кратчайшее расстояние до корневого коммутатора SW1 (обозначим их кружком);

3. Обозначим жирной точкой назначенные порты. Это все порты SW1 и порт 2 SW2;

4. Так как стоимости путей одинаковые у всех портов, то опреде-

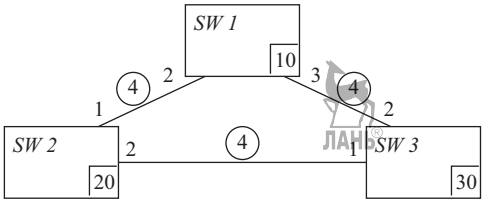


Рис. 3.16. Схема на основе коммутатора с логической петлёй

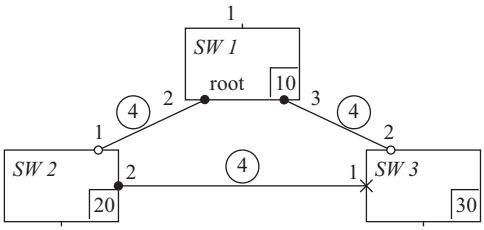


Рис. 3.17. Схема на основе коммутатора без логической петли

ляем резервный порт по приоритетности. Тот порт, где приоритетность больше, переводят в резервное состояние (на рис. 3.17 обозначен крестиком).

Таким образом, образовалось связующее дерево без петель.

Вычисление связующего дерева происходит при включении коммутатора и при изменении топологии. Эти вычисления требуют периодического обмена информацией между коммутаторами связующего дерева, что достигается при помощи специальных пакетов, называемых блоками данных протокола моста (Bridge Protocol Data Unit, BPDU). Пакеты BPDU содержат основную информацию, необходимую для построения топологии сети без петель [12]:

- идентификатор коммутатора, на основании которого выбирается корневой коммутатор;
- расстояние от коммутатора-источника до корневого коммутатора (стоимость корневого маршрута);
- идентификатор порта.

Пакеты BPDU помещаются в поле данных кадров канального уровня, например, кадров Ethernet.

Коммутаторы обмениваются BPDU через равные интервалы времени (обычно 1–4 с). В случае отказа моста (что приводит к изменению топологии) соседние коммутаторы, не получив пакет BPDU в течение заданного времени (Max Age), начинают пересчет связующего дерева [12].

Протокол STP обновлялся несколько раз, последняя его версия описана в документе 802.1D-2004. Эта версия получила название RSTP (Rapid STP, то есть быстрый протокол покрывающего дерева). Новая версия протокола работает значительно быстрее [1].

3.2.5. Маршрутизаторы

Маршрутизаторы — это устройства третьего уровня эталонной модели OSI использующие один или более метрик для определения оптимального пути передачи трафика на основе информации сетевого уровня.

По определению, основное назначение маршрутизаторов — это маршрутизация трафика сети.

Процесс маршрутизации можно разделить на два иерархически связанных уровня [10]:

1. Уровень маршрутизации. На этом уровне происходит работа с таблицей маршрутизации. Таблица маршрутизации служит для определения адреса (сетевого уровня) следующего маршрутизатора или непосредственно получателя по имеющемуся адресу (сетевого уровня) и получателя после определения адреса передачи выбирается определенный выходной физический порт маршрутизатора. Этот процесс на-

зывается *определением маршрута перемещения пакета*. Настройка таблицы маршрутизации ведется протоколами маршрутизации. Этот уровень часто называют уровнем управления (control plain).

2. Уровень передачи пакетов. Перед тем, как передать пакет, необходимо проверить контрольную сумму заголовка пакета, определить адрес (канального уровня) получателя пакета и произвести непосредственно отправку пакета с учетом очередности, фрагментации, фильтрации и т.д. Эти действия выполняются на основании команд, поступающих с уровня маршрутизации [10]. Этот уровень часто называют уровнем данных (data plain).

Основные достоинства маршрутизаторов:

- можно вносить изменения в программную конфигурацию во время работы маршрутизатора, без его перезагрузки и прерывания выполнения сетевых приложений и услуг;
- возможность установки и удаления модулей во время работы маршрутизатора без перезагрузки или выключения системы. Требуется минимальное вмешательство оператора, так как адаптеры порта реконфигурируются автоматически;
- быстрая начальная загрузка (как правило, 35 секунд) обеспечивает быстрый ввод системы в рабочий режим после обновлений операционной системы, сводя к минимуму воздействие на работу сети;
- мониторинг параметров окружающей среды: выдача тревожных сообщений об отклонениях рабочих параметров от нормальных значений;
- самодиагностика и инструментальные средства контроля гарантируют работоспособность модулей перед их включением в работу;
- использование факультативного блока питания повышает отказоустойчивость и позволяет выравнивать нагрузку;
- flash-память обеспечивает быструю, надежную модернизацию программного обеспечения и микрокода с центрального пункта управления сети [1].

Маршрутизаторы обеспечивают сквозную маршрутизацию при прохождении пакетов данных и маршрутизацию трафика между различными сетями на основании информации сетевого протокола или уровня и способны принимать решение о выборе оптимального маршрута движения данных в сети. С помощью маршрутизаторов также может быть решена проблема чрезмерного широковещательного трафика, так как они не переадресовывают дальше широковещательные кадры, если им это не предписано.

Маршрутизаторы и коммутаторы отличаются друг от друга в

нескольких аспектах. Во-первых, коммутаторные соединения осуществляются на канальном уровне, в то время как маршрутизация выполняется на сетевом уровне эталонной модели OSI. Во-вторых, коммутаторы используют физические или MAC-адреса для принятия решения о передаче данных. Маршрутизаторы для принятия решения используют различные схемы адресации, существующие на уровне 3. Они используют адреса сетевого уровня, также называемые логическими или IP-адресами (Internet Protocol).

Структура маршрутизаторов. Структурная схема маршрутизатора представлена на рис. 3.18.

Центральным устройством является процессор, тип которого может различаться в зависимости от класса маршрутизатора, фирмы изготовителя, серии маршрутизатора внутри класса. Основная задача процессора заключается в обработке входящих пакетов для принятия решения об их дальнейшей маршрутизации, при этом скорость, с которой маршрутизатор способен обрабатывать поступающие пакеты, напрямую зависит от типа используемого процессора [10].

Другой важной частью маршрутизатора является его *память*, которая поделена по функциональному признаку:

- *постоянное запоминающее устройство* (ПЗУ, ROM). Используется для хранения загрузочного программного обеспечения, которое запускается первым в момент включения маршрутизатора и в дальнейшем отвечает за его загрузку;
- *flash-память*. Хранит конфигурации операционной системы, которая обеспечивает работу маршрутизатора. Хранит несколько образов операционных систем, чтобы администратор мог перейти к работе на любую из них;
- *память с произвольным доступом* (RAM, ОЗУ). Энергозависимая память, то есть ее содержимое стирается после выключения питания маршрутизатора. Поэтому она используется для хранения промежуточных данных во время работы маршрутизатора;
- *энергонезависимая память* (NVRAM, ППЗУ). Хранятся дополнительные конфигурации маршрутизатора, которые считываются при последующих загрузках.

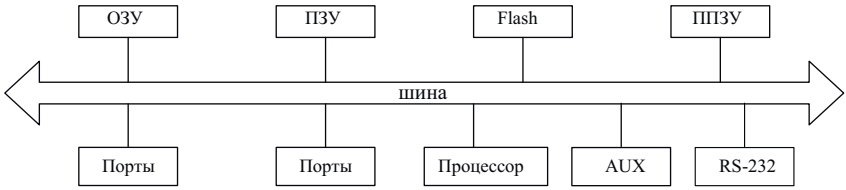


Рис. 3.18. Структурная схема маршрутизатора

Кроме памяти и процессора все маршрутизаторы имеют *интерфейсы*, называемые портами маршрутизатора. Порты обязательно имеют наименование и обязательно пронумерованы, при этом полное имя интерфейса маршрутизатора содержит его тип и номер.

Маршрутизаторы имеют *консольный порт*, предоставляющий асинхронное соединение RS-232. Такой порт позволяет с помощью подключения через консольный кабель управлять маршрутизатором с компьютера.

Также имеется *разъем AUX*, он является вспомогательным и используется для управления маршрутизатором через модем.

Важной составной частью маршрутизатора являются *конфигурированные файлы*. Существует два типа конфигурации: рабочая (или активная) и загрузочная. Рабочая конфигурация располагается в RAM маршрутизатора и определяет его текущие настройки. Загрузочная конфигурация расположена в NVRAM маршрутизатора и содержит команды операционной системы, которые выполняются в момент его загрузки.

Операционная система предназначена для управления активным сетевым оборудованием [10].

Алгоритмы маршрутизации. При разработке алгоритмов маршрутизации часто преследуют одну или несколько из перечисленных ниже целей [10]:

- *оптимальность*. Она характеризует способность алгоритма маршрутизации выбирать «наилучший» маршрут;
- *простота и низкие непроизводительные затраты*. Другими словами, алгоритм маршрутизации должен эффективно обеспечивать свои функциональные возможности с минимальными затратами программного обеспечения и коэффициентом использования;
- *живучесть и стабильность*. Другими словами, они должны четко функционировать в случае неординарных или непредвиденных обстоятельств, таких как отказы аппаратуры, условия высокой нагрузки и некорректные реализации;
- *быстрая сходимость*. Сходимость — это процесс соглашения между всеми маршрутизаторами по оптимальным маршрутам;
- *гибкость*. Другими словами, алгоритмы маршрутизации должны быстро и точно адаптироваться к разнообразным обстоятельствам в сети.

Алгоритмы маршрутизации могут быть классифицированы по типам. Таким образом, алгоритмы маршрутизации бывают:

- *Статическими или динамическими*. Алгоритмы, использующие статические маршруты, просты для разработки и хоро-

шо работают в окружениях, где трафик сети относительно предсказуем, а схема сети относительно проста. Динамические алгоритмы маршрутизации подстраиваются к изменяющимся обстоятельствам сети в масштабе реального времени. Они выполняют это путем анализа поступающих сообщений об обновлении маршрутизации.

- *Одномаршрутными или многомаршрутными*. Некоторые сложные протоколы маршрутизации обеспечивают множество маршрутов к одному и тому же пункту назначения. Такие многомаршрутные алгоритмы делают возможной мультиплексную передачу трафика по многочисленным линиям. Одномаршрутные алгоритмы не могут делать этого. Преимущества многомаршрутных алгоритмов очевидны: они могут обеспечить значительно бóльшую пропускную способность и надежность.
- *Одноуровневыми или иерархическими*. В одноуровневой системе маршрутизации все роутеры равны по отношению друг к другу. В иерархической системе маршрутизации некоторые маршрутизаторы формируют то, что составляет основу (backbone-базу) маршрутизации. Основным преимуществом иерархической маршрутизации является то, что она имитирует организацию большинства компаний и, следовательно, очень хорошо поддерживает их схемы трафика.
- *С интеллектом в главной вычислительной машине или в маршрутизаторе*. В первой системе, рассмотренной выше, интеллект маршрутизации находится в главной вычислительной машине, во втором случае интеллектом маршрутизации наделены маршрутизаторы.
- *Внутридоменными и междоменными*. Некоторые алгоритмы маршрутизации действуют только в пределах доменов; другие — как в пределах доменов, так и между ними. Природа этих двух типов алгоритмов различная. Поэтому понятно, что оптимальный алгоритм внутридоменной маршрутизации не обязательно будет оптимальным алгоритмом междоменной мршрутизации.
- *Алгоритмами состояния канала или вектора расстояний*. Алгоритмы состояния канала (известные также как алгоритмы «первоочередности наикратчайшего маршрута») направляют потоки маршрутной информации во все узлы объединенной сети. Однако каждый маршрутизатор посылает только ту часть маршрутной таблицы, которая описывает состояние его собственных каналов. Алгоритмы вектора расстояния требуют от каждого маршрутизатора посылки всей или части своей

маршрутной таблицы, но только своим соседям. Алгоритмы состояния каналов фактически направляют небольшие корректировки по всем направлениям, в то время как алгоритмы вектора расстояний отсылают более крупные корректировки только в соседние маршрутизаторы [16].

Таким образом, с использованием алгоритмов маршрутизации обеспечивается доставка пакетов по назначению с наибольшей эффективностью. Чаще всего эффективность выражена взвешенной суммой времен доставки сообщений при ограничении на вероятность доставки. Маршрутизация сводится к определению направлений движения пакетов в маршрутизаторах. Выбор одного из возможных в маршрутизаторе направлений зависит от текущей топологии сети (она может меняться хотя бы из-за временного выхода некоторых узлов из строя), длин очередей в узлах коммутации, интенсивности входных потоков и т.п.

3.2.6. Шлюзы

Шлюз — устройство, позволяющее организовать обмен данными между сетевыми объектами, использующими различные протоколы обмена данными. Шлюз выполняет свои функции на уровнях выше сетевого. Он не зависит от используемой передающей среды, но зависит от используемых протоколов обмена данными. Как правило, шлюз выполняет преобразования между какими-либо двумя протоколами (например: NetWare и TCP\IP и т.д.). Некоторые устройства, выполняющие функции шлюза, называют устройствами обслуживания канала. Шлюзы бывают нескольких видов: адресные, протокольные, шлюзы форматов, туннельные шлюзы, шлюзы безопасности (брандмауэры), линейные шлюзы и т.д. Каждый из них имеет свою специфику работы.

Адресные шлюзы соединяют сети с разными пространственными областями каталогов, но с одинаковыми протоколами. Такие шлюзы часто используются, например, в службе обработки сообщений.

Протокольные шлюзы соединяют сети, в которых используются разные протоколы передачи данных. При этом основная задача шлюза заключается в преобразовании протоколов.

Шлюзы форматов объединяют сети, в которых используются разные форматы представления данных (например, символьные кодировки EBCDIC и ASCII). Эти шлюзы выполняют преобразование форматов данных.

Выделяют также *туннельные шлюзы*, использующие для передачи данных через несовместимые области сети относительно несложную методику туннелирования (tunneling). Пакеты данных инкапсулируются в кадры, распознаваемые сетью, через которую планируется

организовать передачу. При этом сохраняются первоначальный формат и разбиение на кадры. На стороне получателя данные восстанавливаются в исходном формате с «отсечением» всей лишней служебной информации.

Шлюзы безопасности (security gateway), или брандмауэры (firewalls), отличаются по специфике решаемых задач от других шлюзов, что позволяет рассмотреть их отдельно. Различают три типа брандмауэров: фильтры пакетов, линейные (circuit) шлюзы и шлюзы приложений.

Наиболее распространенной формой обеспечения безопасности является фильтрация пакетов (packet filtration). Программное обеспечение позволяет анализировать потенциальную опасность каждого пакета по адресам получателя и отправителя, а также по номеру порта.

Процедура фильтрации применима к входящим и/или выходящим пакетам. Реализация этой процедуры на сетевом уровне означает, что стандартная машина (маршрутизатор) будет в состоянии предоставить некоторые функции безопасности всем приложениям, обменивающимся данными через сеть. В результате анализа информации в заголовке пакета на соответствие заранее подготовленному списку приоритетов доступа фильтр пакетов принимает решение «пустить/не пустить». Однако фильтрация имеет следующие потенциально уязвимые места.

Во-первых, происходит снижение производительности шлюза при большом объеме списка привилегий доступа. Второй недостаток заключается в том, что информация заголовка пакета предполагается заведомо правильной, что приводит к успешному использованию фальсифицированной информации заголовка.

Линейные шлюзы (circuit-level gateways) идеальным образом подходят для защиты внешних запросов, сгенерированных в частной безопасной сети. Линейный шлюз функционирует в качестве проводника между инициатором запроса и необходимыми данными, который не подвергает инициатора риску, связанному с передачей данных через небезопасную область сети.

Шлюзы приложений размещают на каждом защищаемом компьютере соответствующее программное обеспечение. Эта более эффективная методика обеспечивает высокий уровень безопасности для работающих в сети компьютеров. Принципиальные ограничения шлюзов приложений связаны с тем обстоятельством, что для каждого работающего в сети компьютера или приложения приходится использовать отдельный шлюз.

3.2.7. Виртуальные локальные сети (Virtual local area Network, VLAN)

Виртуальная сеть образует группу узлов сети, в которой весь трафик, включая и широковещательный, полностью изолирован на

канальном уровне от других узлов сети.

Поэтому VLAN представляет собой логически (программно) обособленный сегмент основной сети. Обмен данными происходит только в пределах одной VLAN. Сетевые устройства в разных VLAN не видят друг друга. Кроме того, широковещательные кадры не могут быть переданы из одной VLAN в другую. Это означает, что передача кадров между разными виртуальными сетями на основании MAC-адреса невозможна независимо от типа адреса — уникального, группового или широковещательного.

В то же время внутри сети кадры передаются по технологии коммутации в соответствии с MAC-адресом, т.е. только на тот порт, который связан с адресом назначения кадра [13]. Физически сеть находится в различных сегментах, но логически связана друг с другом.

Логическое группирование сетевых ресурсов в виртуальных локальных сетях освобождает от ограничений существующей сетевой топологии и кабельной инфраструктуры и упрощает администрирование.

VLAN можно создать только на управляемых устройствах. Одна VLAN может объединять порты нескольких коммутаторов (VLAN с одинаковыми номерами на разных коммутаторах считается одной и той же VLAN).

Можно выделить ряд преимуществ VLAN:

- VLAN является эффективным способом группирования сетевых пользователей в виртуальные рабочие группы, несмотря на их физическое размещение в сети, что обеспечивает гибкость внедрения (см. рис. 3.19);
- VLAN обеспечивают возможность контроля широковещательных сообщений, что увеличивает полосу пропускания, доступную для пользователей;
- VLAN усиливает безопасность сети, определив с помощью фильтров, настроенных на коммутаторе или маршрутизаторе, политику взаимодействия пользователей из разных виртуальных сетей;
- более эффективное использование ресурсов сервера, так как сетевой серверный адаптер с поддержкой VLAN может принадлежать многим VLANs, что уменьшает потребность в маршрутизации трафика к серверу и от него;
- виртуальные сети на основе управляемого программного обеспечения не требуют изменения существующей топологии. Логическое группирование позволяет быстро и легко изменять и реорганизовывать структуру сети с управляющей рабочей станции администратора сети.

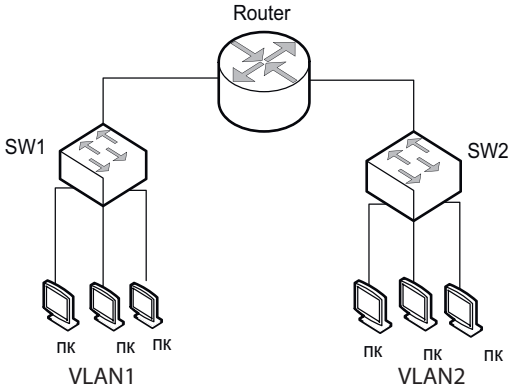


Рис. 3.19. Схема организации VLAN

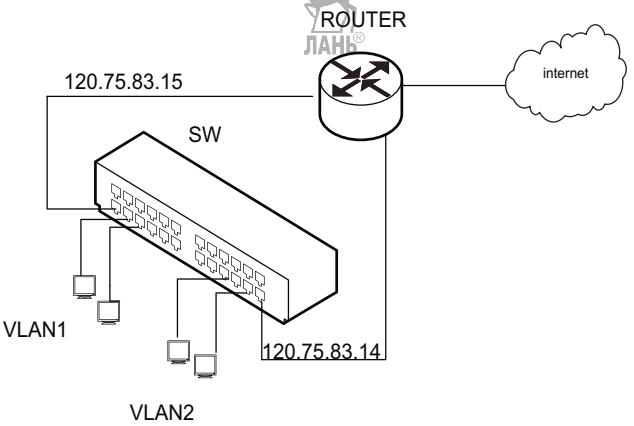


Рис. 3.20. VLAN на базе портов

Объединение сетевых устройств VLAN в отдельные группы может производиться по некоторым общим критериям:

- по виду выполняемых ими функций;
- типу используемых приложений;
- использованию различных видов сетевых ресурсов.

Конфигурация типичной локальной сети определяется физической инфраструктурой соединения устройств, образующих сеть. Группирование пользователей осуществляется исходя из расположения их компьютеров по отношению к коммутатору и основывается на структуре кабелей, ведущих к монтажному шкафу.

Маршрутизатор в виртуальной сети выполняет функцию широковещательного брандмауэра, поддерживает сегментацию сети и защищает ее от атак из магистральной сети. В то же время сегменты, со-

зданные коммутаторами, такими свойствами не обладают. Такой тип сегментации при группировании не учитывает взаимосвязи рабочих групп и требования к ширине полосы пропускания. Вследствие этого они используют один и тот же сегмент и в равной степени претендуют на одну и ту же полосу пропускания, хотя требования к ней для различных групп и подразделений могут значительно различаться.

Существует три основных варианта создания VLAN:

- на основе портов;
- на основе MAC-адресов;
- на основе стандарта IEEE 802.1Q [5].

VLAN на базе портов (port-based) (см. рис. 3.20). Каждый порт назначается в определенную VLAN, независимо от того, какой пользователь или компьютер подключен к данному порту. Это означает, что все пользователи, подключенные к этому порту, будут членами одной VLAN.

Конфигурация портов статическая и может быть изменена только вручную [7, 13].

Основные характеристики:

- применяется в пределах одного коммутатора. Физическое устройство условно делим пополам; получается VLAN1 и VLAN2. Между собой эти пользователи общаться не могут;
- простота настройки. Создание VLAN на основе группирования портов не требует от администратора большого объема ручной работы — достаточно каждому порту, находящемуся в одной VLAN *N* присвоить один и тот же идентификатор VLAN *N* ID;
- возможность изменения логической топологии сети без физического перемещения станции. Достаточно всего лишь изменить настройки порта, например, с VLAN1 на VLAN2, и рабочая станция сразу же получает возможность совместно использовать ресурсы с членами новой VLAN2. Таким образом, VLAN обеспечивает гибкость при перемещениях, изменениях и наращивании сети;
- каждый порт может входить только в один VLAN. Для объединения виртуальных подсетей — как внутри одного коммутатора, так и между двумя коммутаторами, — нужно использовать сетевой уровень. Один из портов каждой VLAN подключается к интерфейсу маршрутизатора, который создает таблицу маршрутизации, для посылки пакетов из одной подсети VLAN1 в другую VLAN2; при этом IP-адреса подсетей должны быть разными.

Недостатком данного решения является то, что один порт каждой VLAN необходимо подключить к маршрутизатору. Это приводит к дополнительным расходам на покупку кабелей и маршрутизатор, при этом порты коммутатора используются весьма расточительно. Эту проблему можно решить при использовании коммутаторов третьего уровня или использовать коммутаторы, которые позволяют включать порт в несколько VLAN [13].

VLAN на базе MAC-адресов (MAC-VLAN) (рис. 3.21). Подключать устройства к той или иной VLAN на основе MAC-адресов. Например, сгруппировать камеры видеонаблюдения, IP-телефоны и т.д. При существовании в сети большого количества узлов этот способ требует выполнения большого количества ручных операций от администратора по маркировке MAC-адресов на каждом коммутаторе. Однако он является более гибким при построении виртуальных сетей на основе нескольких коммутаторов.

Каждый коммутатор поддерживает таблицу MAC-адресов и их соотношение с VLAN. Присвоение отдельных MAC-адресов нескольким VLAN может быть непростой задачей. Это может быть существенным ограничением для совместного использования ресурсов сер-

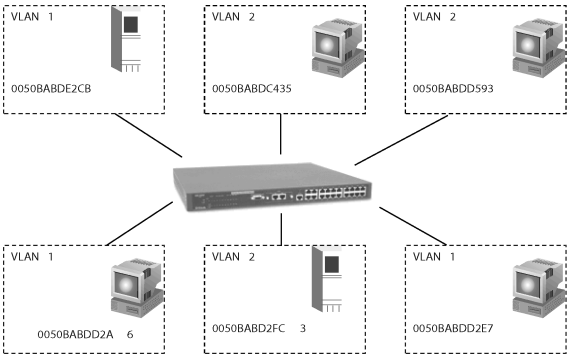


Рис. 3.21. VLAN на базе MAC-адресов

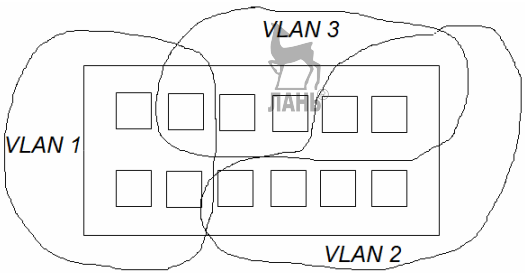


Рис. 3.22. Организация VLAN на базе меток-тегов

вера между несколькими VLAN. Хотя MAC-адрес теоретически может быть присвоен множеству VLAN, это может вызывать серьезные проблемы с существующей маршрутизацией и ошибки, связанные с таблицами пересылки пакетов в коммутаторе.

Широковещательные домены на базе MAC-адресов, позволяют физически перемещать станцию (подключать к любому порту коммутатора), позволяя оставаться ей в одном и том же широковещательном домене без каких-либо изменений в настройках конфигурации.

VLAN на базе меток-тегов. Достоинства данного типа VLAN:

- гибкость и удобство настройки и изменения;
- возможность работы протокола Spanning Tree;
- возможность работы с сетевыми устройствами, которые не распознают метки;
- устройства разных производителей могут работать вместе;
- не нужно применять маршрутизаторы, чтобы связать подсети.

Метод использует дополнительные поля для хранения информации о принадлежности кадра при его перемещениях между коммутаторами сети. IEEE 802.1Q определяет изменения в структуре кадра Ethernet, позволяющая передавать информацию о VLAN по сети (рис. 3.22).

Стандарт IEEE 802.1p специфицирует метод указания приоритета кадра, основанный на использовании новых полей, определенных в стандарте IEEE 802.1Q.

Определенные порты можно добавлять в несколько виртуальных сетей. Для того, чтобы разобраться, где какой порт находится, применяется тегирование. Рассмотрим обычный кадр технологии Ethernet (рис. 3.23).

Если добавляется тег, то он дополняется в дополнительное поле в 4 байта (рис. 3.24).

Максимальный размер кадра 1518 байт при дополнительном поле

Преамбула	АП	АО	L/T	Дата	CRC

Рис. 3.23. Формат кадра Ethernet

АП	АО	TAG	L/T	Данные	CRC

Рис. 3.24. Формат кадра на базе меток-TAG

Рис. 3.25. Поле TAG

получается 1522 байта. Так как кадр длиннее, то снова рассчитывается CRC.

Сеть такой пакет будет выбрасывать, поэтому необходимо ставить конверторы, которые уменьшают размер пакета до 1518 байт за счет поля данных. Четыре байта поля кодируются как показано на рис. 3.25.

Первые 2 байта определяют, что кадр содержит TAG по протоколу IEEE 802.1Q/802.1p.

Последние 2 байта в TAGe содержат следующую информацию:

- 3 бита — приоритет (0–7), который использует стандарт IEEE 802.1p;
- 1 бит Cononical Format Indicator (CFI) зарезервирован для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet;
- 12-битовый идентификатор VLAN — VID — определяет, какой VLAN принадлежит трафик.

Так как поле VID определяет 12 бит, то можно определить 4096 уникальных VLAN. Последний доступный номер 4095 — чёрная дверь или мусорная корзина, куда сбрасываются адреса, которые не присутствуют в сети. Таким образом, количество адресов составляет VID 2–4094.

Способность VLAN 802.1Q добавлять и извлекать метки из заголовков пакетов позволяет VLAN работать с коммутаторами и сетевыми адаптерами серверов и рабочих станций, которые не распознают метки и устройства разных производителей. Поддерживающие этот стандарт устройства могут работать совместно.

Спецификация IEEE 802.1p, создаваемая в рамках процесса стандартизации 802.1Q, определяет метод передачи информации о приоритете сетевого трафика. Стандарт 802.1p специфицирует алгоритм изменения порядка расположения пакетов в очередях, с помощью которого обеспечивается своевременная доставка чувствительного к временным задержкам трафика.

Рис. 3.26. Правила продвижения пакета

Решение о продвижении кадра принимается на основе трёх правил (рис. 3.26):

Правила входящего трафика — относительно принадлежности к VLAN. После того, как кадр принят входным портом коммутатора, решение о его дальнейшей обработке принимается на основании определенных правил, так как принимаемый кадр может относиться как к типу Tagged (тегированных), так и к типу Untagged (нетегированных). В этом случае правилами входного порта определяется, какие типы кадров должны приниматься портом, а какие отфильтровываться. Возможны следующие варианты:

- приём только кадров типа Tagged;
- приём кадров как типа Tagged, так и типа Untagged;
- приём кадров типа Untagged.

По умолчанию для всех коммутаторов правилами входного порта устанавливается возможность приёма кадров обоих типов. Если правилами входного порта определено, что он может принимать кадр Tagged, в котором имеется информация о принадлежности к конкретной VID, то этот кадр передается без изменения. А если определена возможность работы с кадрами типа Untagged, в которой не содержится информация о принадлежности к виртуальной сети, то прежде всего такой кадр преобразуется входным портом коммутатора в тип Tagged (внутри сети все кадры должны иметь метки принадлежности к виртуальной сети) (рис. 3.27).

Чтобы такое было возможно, каждому порту коммутатора присваивается уникальный PVID (Port VILAN Indentifier), определяющий

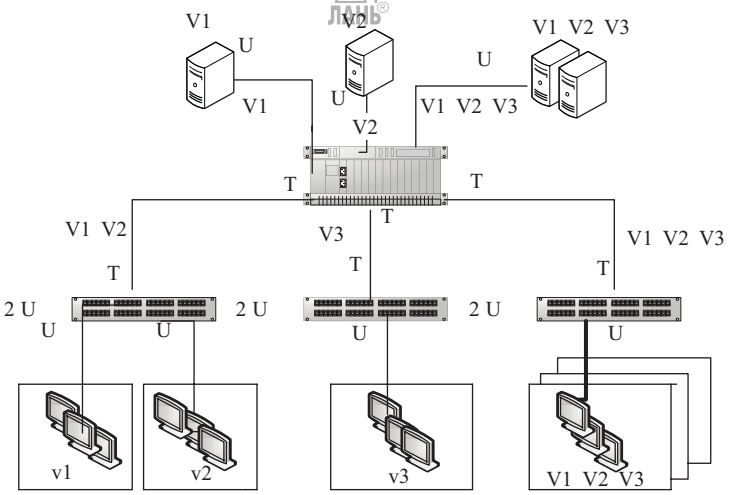


Рис. 3.27. Пример построения VLAN

принадлежность порта к конкретной виртуальной сети внутри коммутатора (по умолчанию все порты коммутатора имеют одинаковый идентификатор PVID = 1).

Все входящие нетегированные кадры (U) автоматически приписываются к той виртуальной сети внутри коммутатора, к которой принадлежит входящий порт, и тегируются (T), что и показано на рис. 3.27.

Правила продвижения между портами. Принимается решение продвигать или отбрасывать кадр. Решение о пересылке принимается на основе таблиц. База фильтрации содержит две таблицы: таблицу MAC (порт, MAC-адрес назначения принятого кадра, TTL — время жизни пакета в сети) и таблицу VLAN (VLAN ID принятого кадра, выходной порт, SVLAN/DVLAN, тип выходного кадра). Кадры после применения входного правила вначале обрабатываются при помощи таблицы MAC, а затем при помощи таблицы VLAN. База фильтрации хранит информацию по VLAN, используемую для коммутации кадров. Она содержит статическую таблицу (Static VLAN, SVLAN) и динамическую таблицу (Dynamic VLAN, DVLAN). Таблица SVLAN ведется вручную администратором. Таблица DVLAN динамически пополняется при помощи протокола GVRP и не может изменяться администратором.

Правила исходящего трафика. Выходное правило решает, должна ли быть информация о признаке VLAN в отправляемом кадре или нет. Решение принимается на основе информации поля Egress Tag Control (контроль признака на выходе) из базы фильтрации. Чтобы попасть в межсетевой маршрутизатор или в оконечную рабочую станцию, кадр должен выйти за пределы коммутируемой сети. Ее выходное устройство «решает», какому порту (или портам) нужно передать пакет и есть ли необходимость удалять из него служебную информацию, предусмотренную стандартом 802.1Q. Дело в том, что традиционные рабочие станции не всегда воспринимают информацию о VLAN по стандарту 802.1Q, но сервер, обслуживающий несколько подсетей с помощью единственного интерфейса, должен ее активно использовать.

Условное деление трафика на внутренний, а также входного и выходного портов позволяет поставщикам нестандартных реализаций VLAN создавать шлюзы для их стыковки с VLAN, соответствующими стандарту 802.1Q.

3.3. Контрольные вопросы

1. Формат кадра Ethernet.
2. Какая сетевая топология применяется в технологии Token Ring?
3. Суть маркерного метода доступа к разделяемой среде.
4. Формат кадра и маркера в технологии Token Ring.
5. Отличие технологии FDDI от Token Ring.
6. Формат кадра FDDI.
7. Структура Fast Ethernet.
8. Формат кадра Fast Ethernet.
9. Как обеспечивается высокая скорость передачи данных в технологии Gigabit Ethernet?
10. Функции концентраторов.
11. Основные типы мостов.
12. Типы коммутаторов. Способы передачи.
13. Какие бывают виды коммутации, их сущность?
14. Принцип и назначение протокола STP.
15. Чем отличается маршрутизатор от коммутатора?
16. Назначение VLAN.
17. На каком уровне эталонной модели работают маршрутизаторы, каковы их функции?
18. Классификация шлюзов, их отличия.
19. Виды VLAN и их особенности.

3.4. Список литературы

1. *Олифер В., Олифер Н.* Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. — 4-е изд. — СПб.: Питер, 2010. — 944 с.
2. *Плешаков В.* CISCO Internetworking Technology Overview. — http://library.tuit.uz/.
3. *Олифер В., Олифер Н.* Базовые технологии локальных сетей. Центр Информационных Технологий. — http://citforum.ru/nets/lvs/glava_5.shtml.
4. http://www.hub.ru/archives/2285.
5. *Куин Л., Рассел Р.* Fast Ethernet. — Киев: BHV, 1998. — 448 с.
6. *Руденков Н.А., Долинер Л.И.* Основы сетевых технологий: учеб. пособие для ВТУЗов. — Екатеринбург: УРФУ, 2011. — 297 с.
7. *Докучаев В.А., Беленькая М.Н., Яковенко Н.В.* Основы сетевых технологий и высокоскоростной передачи данных: учеб. пособие. Часть 1. — 2008. — http://pdst.narod.ru/_60_ElUch/11_ost_wpd_01/index.html.
8. *Филимонов А.* Построение мультисервисных сетей Ethernet. — СПб.: БХВ-Петербург, 2007. — 592 с.

9. *Слепов Н.* 10-гигабитный Ethernet: сегодня и завтра // Первая миля. — 2007. — № 1. — С.10–18.
10. *Кульгин М.* Практика построения компьютерных сетей. Для профессионалов. — М.: Питер, 2000. — 320 с.
11. *Вэк Д., Карнахан Л.* Содержание сети вашей организации в безопасности при работе с Интернетом (Введение в межсетевые экраны (брандмауэры)). Специальная публикация NIST 800-10.
12. Коммутаторы локальных сетей D-Link: учеб. пособие. — 4-е изд. — Москва, 2006. — ebuki.net/?newsid=616.
13. *Смирнова Е.В., Козак П.В.* Технологии современных сетей Ethernet. Методы коммутации и управления потоками данных: учеб. пособие. — СПб.: БХВ-Петербург, 2012. — 272 с.
14. *Будылдина Н.В., Тимченко С.В.* Системы документальной электросвязи: учеб. пособие для вузов. — М.: Горячая линия–Телеком, 2011. — 198 с.
15. *Голощапов С., Шахнович И.* 100GB Ethernet: основные принципы // Первая миля. — 2011. — № 3.
16. *Кульгин М.* Технологии корпоративных сетей. Энциклопедия. — М.: Питер, 2000. — 704 с.
17. IEEE 802.3ba-2010. IEEE Standard for Information Technology. Amendment 4: Media Access Control Parameters, Physical Layers and MaNagement Parameters for 40 Gb/s and 100 Gb/s Operation. — IEEE, 22 June, 2010.
18. *Roese J., Tomizava M., Ishida O.* Optical Transport Network Evolving with 100 Gigabit Ethernet // IEEE Communications Magazine. — March, 2010. — Vol. 50, № 3. — P.s28–s34.
19. *Toyoda H., Ono G., Nishimura Sh.* 100GbE PHY and MAC Layer Implementations // IEEE Communications Magazine. — March, 2010. — Vol. 50, № 3. — P.s41–s47.
20. *D'Ambrosia J.* 100 Gigabit Ethernet and Beyond. — там же, P.6-13.
21. *Cole C., Allouche D., Flens F., Huebner B., Nguyen T.* 100GbE-Optical LAN Technologies // IEEE Communications Magazine Applications Practice. — December 2007. — Vol. 47. — P.12–19.
22. *Слепов Н.* 10-гигабит Ethernet. Часть 2. — http://online.all-over-ip.ru/2012/articles/!/10gigabitnii-ethernet-tss-1-2012.

Глава 4

Протоколы канального уровня

4.1. Основные задачи канального уровня, функции протоколов

Канальный уровень (англ. Data Link layer) — уровень сетевой модели OSI, предназначенный для передачи данных узлам, находящимся в том же сегменте локальной сети.

Задачи канального уровня:

1. Доступ к среде передачи на основе методов доступа , например, с использованием метода доступа CSMA/CD.

2. Управление передачей данных. Структура кадра определена конкретным видом протокола.

Кадр содержит два вида сведений:

- данные пользователя;
- управляющую и служебную информацию.

В свою очередь управляющая информация содержит:

- ограничители и опознаватели различных полей кадра — специальные символы, обычно байты;
- адресные данные (адреса источника и получателя, номер канала и т.д.);
- синхронизирующую информацию, которая предотвращает потерю или дублирование кадра;
- направляющие флаги (управляющие флаги) делят блоки сообщений на информационные и управляющие, определяют порядковый номер блока в сообщении;
- информацию по защите от ошибок (проверочные разряды помехоустойчивого кода, введение обратного канала для запроса и подтверждения о правильности переданного кадра).

3. Реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames).

На канальном уровне обеспечивается проверка каждого передаваемого кадра на корректность его передачи, помещается специальная последовательность бит в начало и конец кадра. На этом уровне вычисляется контрольная сумма, обрабатываются все байты кадра определенным способом и добавляется контрольная сумма к кадру.

На приемном конце станция-получатель снова пересчитывает контрольную сумму полученных данных и сравнивает результат с контрольной суммой полученного кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы

не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например, в Ethernet.

4. Поддержка синхронизации и синфазности. *Синхронизация* — обеспечивает установление и поддержание определенных временны́х соотношений между двумя и более процессами. *Синфазность* — обеспечение циклов фазирования, т.е. разделение кодовых комбинаций или символов первичного алфавита, а также отдельных частей кадра, пакета, фрагмента, блока сообщений.

Цикловое фазирование определяется типом используемого протокола. Существует два класса протоколов:

– бит-ориентированные протоколы;
– байт-ориентированные протоколы.

В байт-ориентированном протоколе нераздельной частью является символ первичного алфавита — байт. Поэтому все служебные признаки могут состоять только из целого числа байт.

В бит-ориентированных протоколах нераздельной частью является бит или единичный элемент. Данные протоколы являются наиболее гибкими и компактными, поэтому нашли более широкое применение.

В общем случае задача циклового фазирования — определение начала и конца каждого отдельного кадра. На практике используются три метода фазирования (кадрирования):

– знаковое кадрирование, когда используются специальные символы (байты) как для указания начала и конца кадра, так и для отделения отдельных полей внутри кадра;
– битовое кадрирование с флагами, когда используется специальная последовательность единичных элементов, называемая флагами, для определения начала и конца кадра.
– кадрирование с указанием длины кадра в поле заголовка.

5. Обеспечение прозрачности передачи. Это относительная независимость функционирования каждого уровня. Одним из условий этой независимости является выполнение передачи данных с более высокого уровня через нижележащий уровень без изменения структуры и формы данных. При передаче данных существует возможность появления в информационном сообщении последовательностей, совпадающих с со служебными символами (флагами). Эта проблема решается с помощью:

– байт-стаффинга (для байт-ориентированных протоколов);
– бит-стаффинга (для бит-ориентированных протоколов).

В байт-ориентированных протоколах каждому служебному символу предшествует специальный символ DLE (Data Link Escape).

Байт-стаффинг заключается в том, что внутри тела кадра (текстовое поле) к каждому символу DLE, по какой-либо причине присутствовавшему в тексте, добавляется ещё такой же символ DLE. Вставка добавочного символа выполняется при вводе данных в информационное поле.

В бит-ориентированных протоколах вводятся специальные символы, чтобы флаг не был ошибочно обнаружен в информационном теле кадра, если в текстовом поле встретится подобная последовательность. Для исключения такой возможности используют процедуру бит-стаффинга.

6. Оптимальное использование связных ресурсов. Протоколы управления физическим каналом должны обеспечить эффективное использование пропускной способности дискретного канала. Это касается следующих моментов:

- направленность передачи данных (симплексная, дуплексная, полудуплексная);
- объем служебной информации (специальные управляющие символы);
- избыточная информация для обнаружения и исправления ошибок;
- зависимость от передающей среды.

Протоколы канального уровня должны допускать различные способы организации канала. В качестве передающей среды могут использоваться:

- телефонные каналы (коммутируемые и некоммутируемые);
- коаксиальный кабель;
- оптический кабель;
- спутниковые и радиорелейные каналы.

В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей — именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN [3].

В локальных сетях протоколы канального уровня используются

компьютерами, мостами, коммутаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

В глобальных сетях канальный уровень часто обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка–точка» (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B. В таких случаях для доставки сообщений между конечными узлами через всю сеть используются средства сетевого уровня. Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и Frame Relay.

В целом канальный уровень представляет собой практически законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня оказываются транспортными средствами и могут допускать работу поверх них непосредственно протоколов прикладного уровня или приложений без привлечения средств сетевого и транспортного уровней. Например, существует реализация протокола управления сетью SNMP непосредственно поверх Ethernet, хотя стандартно этот протокол работает поверх сетевого протокола IP и транспортного протокола TCP. В этом случае применение такой реализации будет ограниченным. Такая реализация и не подходит для составных сетей разных технологий, например Ethernet и X.25. Также не подходит для тех сетей, в которых во всех сегментах применяется Ethernet, но между сегментами существуют петлевидные связи. В двухсегментной сети Ethernet, объединенной мостом, реализация SNMP над канальным уровнем возможна.

Тем не менее, для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня — сетевой и транспортный [1–3].

Наиболее существенными характеристиками метода передачи, а значит, и протокола, работающего на канальном уровне, являются следующие [3]:

- асинхронный / синхронный;
- символьно-ориентированный / бит-ориентированный;
- с предварительным установлением соединения / дейтаграммный;

- с обнаружением искаженных данных / без обнаружения;
- с обнаружением потерянных данных / без обнаружения;
- с восстановлением искаженных и потерянных данных / без восстановления;
- с поддержкой динамической компрессии данных / без поддержки.

4.2. Байт-ориентированные протоколы

Байт-ориентированный протокол BSC. Байт-ориентированный протокол (или асинхронный протокол) обеспечивает передачу сообщения по информационному каналу в виде последовательности байтов.

Наиболее известным байт-ориентированным протоколом является BSC (Binary Synchronous Communication) — протокол двоичной синхронной связи. Байт-ориентированный протокол BSC разработан фирмой IBM [1]. Формат кадра BSC байт-ориентированнго протокола с синхронной передачей сигналов приведен на рис. 4.1.

1 байт	1 байт	1 байт		1 байт		1 байт	1 байт
SYN	SYN	SOH	Заголовок	STX	Поле данных	ETB/ETX	FCS

Рис. 4.1. Формат кадра BSC байт-ориентированного протокола с *синхронной* передачей сигналов

На рис. 4.1:

- SYN (Synchronous Idle) — синхросимвол, в коде ASCII символ SYN равен 0010110, а в коде EBCDIC — 00110010;
- SOH (Start of Heading) — начало заголовка (НЗ);
- STX (Start of TeXt, ASCII 0000010) — начало текста (НТ);
- ETX (End of TeXt, ASCII 0000011) — конец текста (КТ);
- ETB (End of Block) — конец блока (КБ);
- FCS (Frame Control Sequence) — контрольная сумма.

Синхронизация достигается за счет того, что передатчик перед каждым блоком символов добавляет два или более управляющих символа, называемых символами SYN. Символы SYN выполняют две функции: во-первых, они обеспечивают приемнику побитную синхронизацию; во-вторых, как только битовая синхронизация достигается, они позволяют приемнику начать распознавание границ символов SYN, т.е. обеспечивают байтовую синхронизацию. После того, как приемник начал отделять один символ от другого, можно задавать границы начала кадра с помощью другого специального символа. Обычно в символьных протоколах для этих целей используется символ начала текста STX. Другой символ отмечает окончание кадра — ETX.

Однако такой простой способ выделения начала и конца кадра хорошо работает только в том случае, если внутри кадра нет символов

STX и ETX. При подключении к компьютеру алфавитно-цифровых терминалов такая проблема действительно не возникает. Но когда синхронные байт-ориентированные протоколы стали использовать и для связи компьютера с компьютером, то в этом случае данные внутри кадра могут быть любые, например, если между компьютерами передается программа. Для таких передач были разработаны «прозрачные» протоколы.

Прозрачность достигается за счет того, что перед управляющими символами STX и ETX всегда вставлялся символ DLE (Data Link Escape). Такая процедура называется *байт-стаффингом* (stuff — вставка, заполнитель). А если в поле данных кадра встречалась последовательность DLE ETX, то передатчик удваивал символ DLE, то есть порождал последовательность DLE DLE ETX. Приемник, встретив подряд два символа DLE DLE, в том числе и в поле данных, всегда удалял первый, но оставшийся DLE уже не рассматривал как начало управляющей последовательности, то есть оставшиеся символы DLE ETX считал просто пользовательскими данными.

Контрольная последовательность FCS вычисляется путем суммирования всех байт кадра, начиная с поля «заголовок», и служит она для обнаружения ошибок. Принятая в составе кадра и посчитанная на приемной стороне контрольные суммы должны совпадать, в противном случае кадр считается принятым неверно. Кадр, в котором обнаружена ошибка, бракуется и на передающую сторону направляется запрос на повторную передачу кадра.

Таким образом, повышение достоверности обеспечивается применением корректирующего кода и решающей обратной связи с ожиданием (РОС-ОЖ).

Отличительными чертами протокола BSC являются следующие:

- возможность представления информации 8-битным расширенным двоичным кодом EBCDIC;
- применение для обнаружения ошибок помехоустойчивого циклического кода с контрольной последовательностью CRC-16;
- обеспечение прозрачности по коду [2].

Байт-ориентированный протокол DDCMP. Байт-ориентированный протокол DDCMP (Digital Data Communication Message Protocol) разработан в фирме Digital Equipment Corporation (DEC). Формат заголовка кадра протокола приведен на рис. 4.2

DDCMP предназначен для обеспечения синхронной работы по дуплексным и полудуплексным соединениям, устанавливаемым по коммутируемым или выделенным каналам, в сетях «от точки к точке» или многоточечным соединениям. Причем, в последнем случае одна станция является первичной (основной), а другие — вторичными

(ведомыми).

Перед началом передачи любая из станций должна послать «запрос» и получить на него «подтверждение», после чего информация передается в виде нумерованных блоков, т.е. каждый передаваемый блок имеет свой номер.

Протокол предусматривает подтверждение 255 ранее принятых пронумерованных сообщений одной операцией [3].

Рис. 4.2. Формат кадра DDCMP

На рис. 4.2 SYN (Synchronous Idle) — синхросимвол, в коде ASCII символ SYN равен 0010110, а в коде EBCDIC — 00110010, предназначен для синхронизации устройств;

DDCMP обеспечивает синхронизацию по кадрам и сообщениям (предполагается, что побитовая синхронизация обеспечивается на физическом уровне).

Синхронизация реализуется с помощью стартстопного метода передачи, в качестве стартовой посылки используются 2 байта синхронизации (SYN), посылаемые в начале каждого кадра.

В формате кадра этого протокола выделено две области:

– область управления;

– информационная область.

Информационный кадр отличается от управляющего наличием в заголовке кадра символа SOH — начало заголовка, если вместо SOH передается ENQ (Enquiry) — «кто там?», то кадр считается не информационным, а управляющим.

Управляющие сообщения имеют фиксированную длину, а сообщения, переносящие данные, — переменную длину. Размер передаваемого сообщения, переносящего данные, указывается в специальном поле этого сообщения. Процедуры передачи информации возможны как в асинхронном, так и в синхронном режимах.

Когда протокол DDCMP применяется в целях обслуживания многоточечной линии, то для организации взаимодействия используется пелингование подключенных к линии станций. В случае полудуплексной линии («точка–точка») для указания передающей стороны используется специальный бит «выбора» в заголовке сообщения протокола DDCMP. Бит «выбор» также используется первичной станцией для того, чтобы информировать вторичную станцию о том, что она может выполнить передачу данных. Вторичные станции не имеют возможности передавать данные непосредственно друг другу, все взаимодействия выполняются только с помощью первичной станции.

Счетчик — фиксирует длину передаваемого сообщения. Благода-

ря наличию поля счетчика в заголовке, передатчик может формировать кадры произвольной длины.

Ответ — с тем, чтобы определить все ли сообщения будут доставлены, в протоколе DDCMP применяется механизм конвейера (pipeline). Сообщениям назначаются последовательные номера. При этом подтверждается прием всех сообщений вплоть до указанного последовательного номера. Подтверждение может быть передано и в сообщениях, переносящих данные.

ПН — последовательный номер сообщения. Перед началом передачи любая из станций должна послать «запрос» и получить на него «подтверждение», после чего информация передается в виде нумерованных блоков, т.е. каждый передаваемый блок имеет свой номер.

Для выявления ошибок используются две контрольные суммы (CRC). Одна контрольная сумма (CRC1) вычисляется для передаваемых данных, другая (CRC2) — для заголовка. Выявление ошибок влечет за собой посылку сообщения с признаком NAK (Negative Acknowledgment-НЕТ) в передающий узел, при этом указывается также последовательный номер последнего правильно принятого сообщения. Ошибочное сообщение с целью повторной передачи ставится в очередь готовых для передачи сообщений. Если в течение некоторого времени не получен положительный ответ от приемника, то производится повторная передача предыдущего блока.

Протокол DDCMP предусматривает работу в четырехпроводном режиме: по прямому каналу передается информация, по обратному — сигналы подтверждения правильного приема кадров [3].

DDCMP является кодонезависимым: отсутствуют ограничения на любые комбинации бит и байт в информационной области. Кодонезависимость обеспечивается подсчетом числа байт в информационной области и передачей его в заголовке информационного кадра.

4.3. Бит-ориентированные протоколы

4.3.1. Протокол канального уровня HDLC (High-Level Data Link Control)

Протокол HDLC — это протокол канального уровня, который обеспечивает передачу последовательности пакетов через физический канал, искажения в котором вызывают ошибки в передаваемых данных, потерю, дублирование пакетов и нарушения порядка прибытия пакетов к адресату.

Протокол HDLC является базовым для целой группы протоколов канального уровня, используемых как в глобальных, так и в локальных компьютерных сетях (рис. 4.3):

– LAPB (Link Access Procedure Balanced) — сбалансированная

процедура доступа к звену передачи данных (применяется в стандарте X.25);

- LAPD (Link Access Procedure D-channel) — предназначен для управления звеном в цифровых сетях с интеграцией служб ISDN (Integrated Services Digital Network) — система, в которой по телефонным каналам передаются только цифровые сигналы, в том числе и по абонентским линиям, то есть конечный абонент передает данные непосредственно в цифровой форме;
- LLC (Logical Link Control) — управление логическим каналом;
- LAPM (Link Access Protocol for Modems) — протокол коррекции ошибок в стандарте V.42 для модемов, предназначен для коммутируемой телефонной сети PSTN (Public Switched Telephone Network);
- Frame Relay — представляет собой стандартный протокол объединения локальных сетей, который обеспечивает методы быстрой и эффективной передачи информации от пользовательских устройств до мостов и маршрутизаторов ЛВС;
- PPP (Point-to-Point Protocol) — обеспечивает связь с удаленными сетями через стандартный PPP-сервер. Протокол PPP также позволяет серверу удаленного доступа принимать входящие вызовы от программ удаленного доступа других разработчиков, поддерживающих PPP;
- SDLC (Synchronous Data Link Control) — синхронное управление звеном данных, разработан компанией IBM для системной сетевой архитектуры SNA;
- LAPX (расширенный LAPB) — используется в терминальных системах и в стандарте телетекса. Является полудуплексным вариантом HDLC.

Структура кадра HDLC представлена на рис. 4.4.

Единица данных, передаваемая как целое через информационный канал, называется кадром.

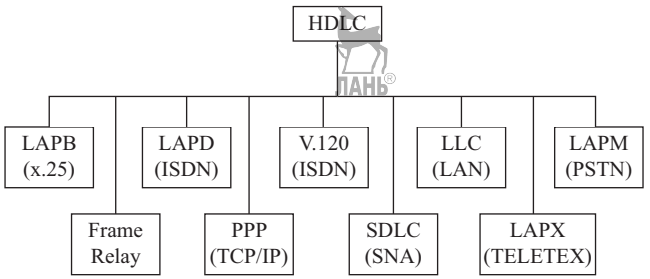


Рис. 4.3. Семейство протокола HDLC

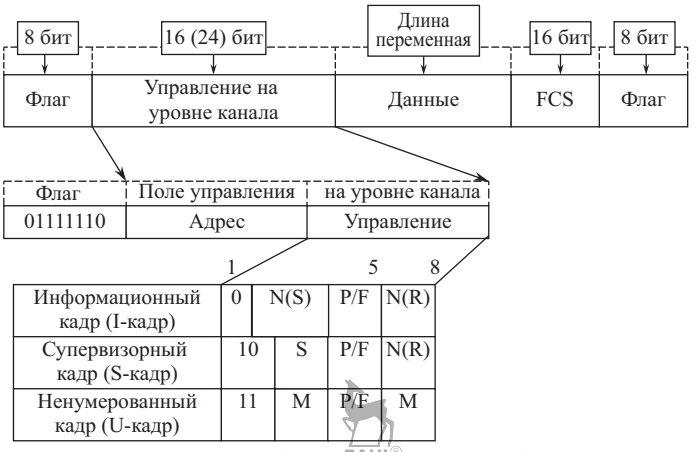


Рис. 4.4. Структура кадра HDLC

Флаг. Все кадры должны начинаться и заканчиваться полями флага 01111110. Станции, подключенные к каналу, постоянно контролируют двоичную последовательность флага. Флаги могут постоянно передаваться по каналу между кадрами HDLC. Для индексации исключительной ситуации в канале могут быть посланы семь подряд идущих единиц. Пятнадцать или большее число единиц поддерживают канал в состоянии покоя. Если принимающая станция обнаружит последовательность битов, являющихся флагом, она тем самым уведомляется о начале кадра, об исключительной (с аварийным завершением) ситуации или ситуации покоя канала. При обнаружении следующей флаговой последовательности станция будет знать, что поступил полный кадр.

Адресное поле определяет первичную или вторичную станции, участвующие в передаче конкретного кадра. Каждой станции присваивается уникальный адрес.

В несбалансированной системе адресные поля в командах и ответах содержат адрес вторичной станции. В сбалансированных конфигурациях командный кадр содержит адрес получателя, а кадр ответа содержит адрес передающей станции.

Управляющее поле задает тип команды или ответа, а также порядковые номера, используемые для отчетности о прохождении данных в канале между первичной и вторичной станциями. Формат и содержание управляющего поля (рис. 4.4) определяют кадры трех типов: информационные (I), супервизорные (S) и ненумерованные (U).

I-кадры (информационные) служат для переноса самой информации или данных.

S-кадры (супервизорные) используются для восстановления кад-

ров, потерянных из-за искажений в канале, а также для управления потоками данных.

U-кадры (ненумерованные) используются для установления соединения и разъединения, завершения соответствующих режимов канала передачи.

Управление потоком в HDLC осуществляется при помощи передающих и принимающих окон. Окно устанавливается на каждом конце канала связи, чтобы обеспечить резервирование ресурсов (ресурсы вычислителя, пространство буфера) обоих станций.

Окна в принимающем и передающем узлах управляются переменными состояния, которые представляют собой состояние счетчика. Передающий узел поддерживает переменную состояния N(S) — порядковый номер передаваемого — следующего по очереди I-кадра. Принимающий узел поддерживает переменную состояния приема N(R) или номер по порядку ожидаемого кадра от противоположной станции.

P/F — бит запроса и завершения.

Если P=1, то происходит запрос на подтверждение кадра и станции необходимо ответить супервизорным кадром, что информационный кадр доставлен или нет.

Если F=1 — завершение передачи.

Так как протокол HDLC был разработан для управления звеном данных, то для начальной установки звена данных выбираются специальные режимы, из которых наиболее распространены:

- *режим нормального ответа* (РНО), где первичная станция (ПС) может передавать данные, вторичная станция (ВС) хранит режим молчания до тех пор, пока не получит запрос от вторичной станции;
- *режим асинхронного ответа* (РАО) позволяет вторичной станции инициировать передачу без получения явного разрешения от первичной станции (обычно, когда канал свободен, в состоянии покоя). Этот режим придает большую гибкость работы вторичной станции. Могут передаваться один или несколько кадров данных или управляющая информация, отражающая изменение статуса вторичной станции. РАО может уменьшить накладные расходы, поскольку вторичная станция, чтобы передать данные, не нуждается в последовательности опроса. Как правило, такой режим используется для управления соединенными в кольцо станциями или же в многоточечных соединениях с опросом по цепочке. В обоих случаях вторичная станция может получить разрешение от другой вторичной станции и в ответ на него начать передачу.

Таким образом, разрешение на работу продвигается по кольцу или вдоль соединения;

– *асинхронно сбалансированный режим* (АСР) используют комбинированные станции. Комбинированная станция может инициировать передачу без получения предварительного разрешения от другой комбинированной станции. Этот режим обеспечивает двусторонний обмен потоками данных между станциями и является основным (рабочим) и наиболее часто используемым на практике.

От того, какой режим используется (РНО или АСР), зависит содержимое адресного поля. Адресное поле кадра содержит адрес либо ООД (оконечное оборудование данных), либо АПД (аппаратура передачи данных) центра коммутации пакетов. Если кадр является командным, то формируется адрес получателя, если же кадр ответный, то формируется адрес отправителя. В РНО адрес всегда относится к вторичной станции, то есть он не несет адреса принимаемой станции.

Процедуры управления канального уровня обеспечивают прозрачность канала за счет бит-стаффинга (вставкой нулевых битов).

Протокол HDLC является бит-ориентированным. В нем как управляющие сообщения, так и сообщения с данными переносятся в блоках стандартного формата, называемых кадрами. При передаче данных формируется проверочная последовательность битов (два октета), которая включается в кадр. При приеме кадра повторно формируется проверочная последовательность битов, которая сравнивается с принятой. Если обе совпадают, то принятый кадр считается корректным. В противном случае фиксируется искажение принятого кадра. При искажении флагов, разделяющих последовательно передаваемые кадры, два кадра сливаются в один искаженный кадр. Процедура формирования проверочных последовательностей битов при передаче и приеме гарантирует обнаружение искажений этого типа [4].

Форматы информационного, супервизорного и ненумерованного кадров приведены на рис. 4.4. Первый бит информационного кадра равен 0 — идентификатор I-кадра.

В байте управления супервизорного кадра указывается тип команды ответа S:

– ГП (готов к приему), то есть выдающая этот кадр станция готова к приему следующего кадра. Используется в режимах РНО и АСР;
– НГП (получатель не готов к приему), то есть станция временно не может принимать I-кадры и запрет остается в силе до посылки кадра ГП. Используется в режимах РНО и АСР;
– ОТК (отказ). Работает только в режиме АСР и означает за-

прос повторной передачи всех I-кадров, начиная с того номера, где произошла ошибка;

– ВОТК (выборочный отказ). Для АСР позволяет запросить повторную передачу только первого одного кадра с номером N(R) — по порядку искаженного кадра.

В байте управления информационного кадра (I-кадра) указываются номера по порядку передаваемого кадра $N(S) = 0, 1, ..., 7$ и номера по порядку ожидаемого кадра от противоположной станции $N(R) = 0, 1, ..., 7$.

В супервизорных кадрах указывается только номер N(R) искаженного кадра.

Поле М ненумерованного кадра используется для специфического типа кадра (биты функции модификатора). Здесь записываются команды кадра:

– УРНО — установить режим нормального ответа;
– УАСР — установить АСР;
– РЗД — разъединить;
– КО — кадр отвергнут и т.д.

Битовое поле модификатора М в ненумерованных кадрах позволяет определить до 32 разновидностей кадров. Двадцать из них определены и предназначены главным образом для запуска и завершения процедур на уровне канала, а также для передачи информации о состоянии выполнения этих процедур.

U-кадры позволяют установить логическую связь между первичной и вторичной станциями, установить режим функционирования между ними, также используются для целей управления: инициализации или разъединения, тестирования, сброса и идентификации станции и т.д.

Поле FCS (контрольная последовательность кадра) используется для обнаружения ошибок передачи между двумя станциями. Передающая станция осуществляет вычисления над потоком данных пользователя, и результат этого вычисления включается в кадр в качестве поля FCS. В свою очередь, принимающая станция производит аналогичные вычисления и сравнивает полученный результат с полем FCS. Если имеет место совпадение, велика вероятность того, что передача произошла без ошибок. В случае несовпадения принимающая станция посылает отрицательное подтверждение, означающее, что необходимо повторить передачу кадра. Вычисление FCS называется циклическим контролем по избыточности и использует производящий полином в соответствии с рекомендацией MKKTT V.41 [4].

Режим нормального ответа (РНО). В режиме нормального ответа связь может быть как двухточечной, так и многоточечной, но

в последнем случае допускается только одна главная станция (первичная, ПС), остальные являются вторичными (ВС). В этом режиме вторичная станция может начать передачу только после разрешения от первичной станции. Режим нормального ответа называется несбалансированным режимом работы.

Рассмотрим пример работы протокола в режиме РНО (рис. 4.5).

Режим нормального ответа (РНО). Первичная станция передает 13 кадров. Окно передач от 0 до 7. Ошибки в 4 и в 7 кадрах. Запрос через окно передач. Ошибка в 4-ом кадре означает неправильный прием кадра номер 3, т.к. счетчик считает количество кадров, а нумерация кадров начинается с нуля.

Прежде чем передать какую-либо информацию между ПС и ВС, должно быть установлено логическое соединение. Это достигается обменом двумя ненумерованными кадрами. ПС посылает ненумерованный кадр для установления режима нормального ответа, в котором бит опроса равен единице (P=1), а в поле адреса стоит ее собственный адрес. В ответ ВС посылает ненумерованный кадр, в котором бит ответа равен единице (F=1). Процедура установления соединения включает также инициализацию идентифицирующих переменных

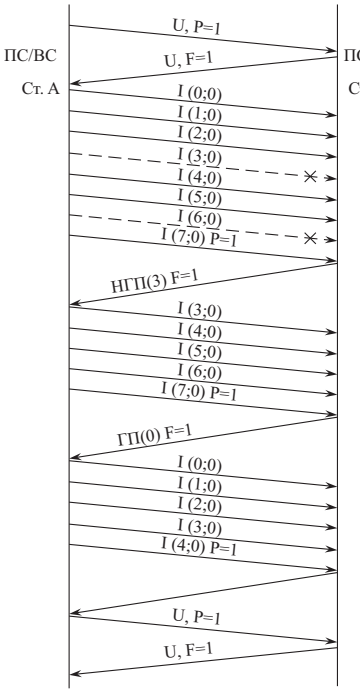


Рис. 4.5. Режим РНО

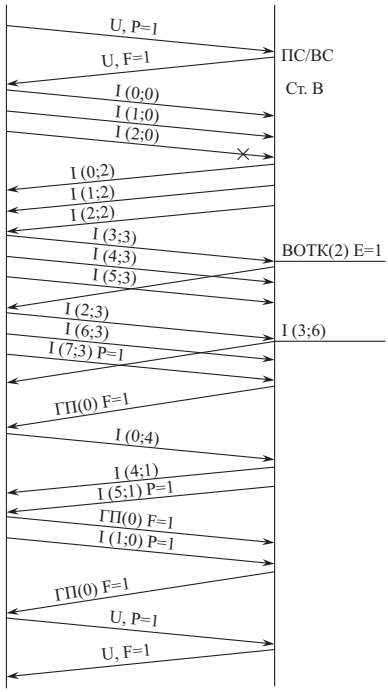


Рис. 4.6. Режим АСР

каждой станции. Эти переменные используются в процедурах управления ошибками и потоком.

После установления режима ПС начинает передачу данных. Получив кадр, ВС производит проверку наличия ошибок в нем и сравнивает номер кадра N(S) с тем номером N(R), который она ожидает; если кадр был принят верно, то N(R) увеличивается на 1. Когда окно передачи завершено, то в 7 по номеру кадре происходит запрос на подтверждение кадров, где P=1. Если кадры приняты правильно и N(S)=N(R), то ВС сформирует ответ ГП с номером следующего кадра, который ожидает ВС, и с битом окончания F=1. ПС, получив такой ответ, продолжит передачу данных. Если кадр принят с ошибкой или N(S)≠N(R), то на запрос от ПС будет выдан ответ НГП с номером ошибочного кадра. Получив такой ответ, ПС должна повторить передачу кадров, начиная с ошибочного.

В заключение после передачи всех данных ПС прерывает связь, посылая для этого кадр РЗД (разъединить — ненумерованным кадром, U-кадр P=1) и получая от ВС ответный кадр НИЗ (ненумерованное извещение, U-кадр F=1).

Асинхронно-сбалансированный режим (АСР). Такой режим используется в сетях, когда обе станции имеют равные права и каждая реализует функции как первичной, так и вторичной станции. Используется главным образом для двухточечных звеньев компьютерных сетей при дуплексной передаче.

Рассмотрим пример работы протокола в режиме АСР (рис. 4.6).

Например, от станции А передается 10 кадров в 3-м кадре ошибка. Окно передач от 0 до 7. От станции В передается 6 кадров. Ошибок нет. Запрос через окно передач от обеих станций.

После установления соединения ПС и ВС могут начать передачу кадров в любое время и при этом каждая станция будет проверять принимаемые от другой станции кадры на наличие ошибок. Если кадры принимаются верно, то на запрос идет подтверждение ГП. В нашем примере в 3 кадре произошла ошибка. В этом случае станция В определяет, что после первого кадра по номеру, правильно принят 4, а третий искажен. В этом случае станция В отправляет супервизорный кадр, выборочный отказ «ВОТК». Станция А, получив выборочный отказ от 2-го кадра по номеру, направляет его повторно.

4.3.2. Протокол кадра SLIP (Serial Line Internet Protocol)

Протокол SLIP позволяет в потоке передаваемых бит, которые поступают по выделенному (или коммутируемому) каналу, распознать начало и конец IP-пакета. Помимо протокола IP, другие протоколы сетевого уровня SLIP не поддерживает.

Чтобы распознать границы IP-пакетов, протокол SLIP предусмат-

ривает использование специального символа END, значение которого в шестнадцатеричном представлении равно C0. Применение специального символа может породить конфликт: если байт пересылаемых данных тождественен символу END, то он будет ошибочно определен как признак конца пакета. Чтобы предотвратить такую ситуацию, байт данных со значением, равным значению символа END, заменяется составной двухбайтовой последовательностью, состоящей из специального символа «ESC» («DB» — 11011011) и кода DC (11011100) [3].

Если же байт данных имеет тот же код, что и символ SLIP ESC, то он заменяется двухбайтовой последовательностью, состоящей из собственно символа SLIP ESC и кода DD (11011101). После последнего байта пакета IP передается символ «END» [3].

Механизм формирования составных последовательностей показан на рис. 4.7. Здесь приведен стандартный IP-пакет (один байт которого тождественен символу END, а другой — символу SLIP ESC) и соответствующий ему SLIP-пакет, который больше на 4 байта.

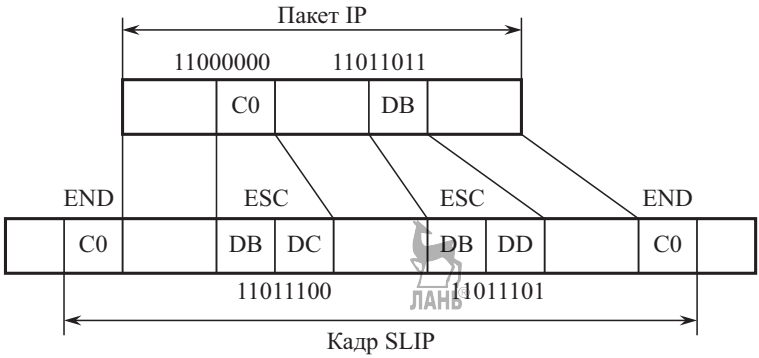


Рис. 4.7. Соответствие между кадром SLIP и пакетом IP

Вставка символа END перед началом кадра позволяет принимающей стороне избавиться от любой помехи на линии связи, но протокол возлагает задачу по определению и исправлению пакетов данных и сообщений полностью на вышележащие протоколы, то есть на сетевой и транспортный уровень TCP/IP.

Хотя в спецификации протокола SLIP не определена максимальная длина передаваемого пакета, реальный размер IP-пакета не должен превышать 1006 байт. Данное ограничение связано с первой реализацией протокола SLIP в соответствующем драйвере для Berkley Unix, и его соблюдение необходимо для поддержки совместимости разной реализации SLIP (большинство современных реализаций позволяют администратору самому установить размер пакета, а по умолчанию используют размер 1500 байт).

Программа управления SLIP загружается и выгружается по мере надобности. Большинство программ управления SLIP имеют возможность набирать телефонный номер провайдера. Программное обеспечение, реализующее работу с протоколом SLIP (TCP-manager), выполняет функции управления сетевым устройством, то есть является драйвером сетевого устройства, такого, как модем.

Сетевое устройство принимает IP-пакеты от программы, посылающей их, обкладывает своей служебной информацией и передаёт устройству последовательной передачи данных (модему, в последовательный порт и т.п.). На другом конце линии аналогичная программа принимает символы, приходящие с устройства последовательной передачи данных, освобождает от служебной информации и передаёт то, что получилось, а должны получаться при этом IP-пакеты, соответствующей программе (сетевого уровня), которая обрабатывает IP-пакеты [3].

Для установления связи по протоколу SLIP компьютеры должны иметь информацию об IP-адресах друг друга. Однако возможна ситуация, когда при осуществлении соединения между хостом и маршрутизатором последнему понадобится передать хосту информацию о его IP-адресе. В протоколе SLIP нет механизмов, дающих возможность обмениваться адресной информацией.

Недостатки SLIP:

- не обеспечивает обмен адресной информацией, что не позволяет использовать SLIP для некоторых видов сетевых услуг. Если провайдер использует динамическое присвоение IP-адресов, то при каждом новом соединении компьютер будет получать новый IP-адрес. Следовательно, другие компьютеры в сети будут вынуждены искать его под неизвестно каким адресом;
- отсутствие индикации типа протокола, пакет которого «вкладывается» в кадр SLIP. Поэтому через последовательную линию по протоколу SLIP можно передавать трафик лишь одного сетевого протокола — IP. Эти функции обеспечивают либо вышележащие протоколы (IP, UDP или TCP), либо нижележащие протоколы;
- не предусмотрены процедуры обнаружения и коррекции ошибок. Эти функции обеспечивают протоколы вышележащих уровней (IP, UDP, TCP);
- низкая пропускная способность линии связи вынуждает сокращать время передачи пакетов, уменьшая объем содержащейся в них служебной информации. Эта задача решается с помощью протокола CSLIP (Compressed SLIP), поддерживающего сжатие заголовков пакетов [3, 5].

Появление CSLIP объясняется тем фактом, что при использовании программ типа Telnet, Rlogin и других для пересылки одного байта данных требуется переслать 20-байтовый заголовок пакета IP и 20-байтовый заголовок пакета TCP (итого 40 байтов). Спецификация CSLIP обеспечивает сжатие 40-байтового заголовка до 3–5 байтов. На сегодняшний момент большинство реализаций протокола SLIP поддерживают спецификацию CSLIP.

Таким образом, протокол SLIP выделяет последовательность байтов, формирующих пакет IP, и ничего более. Он не имеет механизмов передачи адресной информации, идентификации типа протокола сетевого уровня, определения и коррекции ошибок. Но он очень прост, что обеспечивает легкость его реализации [5].

4.3.3. Протокол PPP (Point-to-Point Protocol — протокол двухточечной связи)

Протокол PPP (Point-to-Point Protocol — протокол двухточечной связи) является стандартным протоколом Интернета. Протокол PPP так же, как и HDLC, представляет собой целое семейство протоколов, в которое, в частности, входят [3]:

- протокол управления линией связи (Link Control Protocol, LCP);
- протокол управления сетью (Network Control Protocol, NCP);
- многоканальный протокол PPP (Multi Link PPP, MLPPP);
- протокол аутентификации по паролю (Password Authentication Protocol, PAP);
- протокол аутентификации по квитированию вызова (Challenge Handshake Authentication Protocol, CHAP).

При разработке протокола PPP за основу был взят формат кадров HDLC и дополнен несколькими полями. Эти дополнительные поля протокола PPP вложены в поле данных кадра HDLC. Позже были разработаны стандарты, описывающие вложение кадра PPP в кадры Frame Relay и других протоколов глобальных сетей. Хотя протокол PPP и работает с кадром HDLC, он не поддерживает, подобно стандартной версии протокола HDLC, процедуры надежной передачи кадров и управления их потоком.

Особенностью протокола PPP, отличающей его от других протоколов канального уровня, является *сложная переговорная процедура* принятия параметров соединения. Стороны обмениваются различными параметрами, такими как качество линии, размер кадров, тип протокола аутентификации и тип инкапсулируемых протоколов сетевого уровня.

В корпоративной сети конечные системы часто отличаются размерами буферов для временного хранения пакетов, ограничениями на

размер пакета, списком поддерживаемых протоколов сетевого уровня. Скорость передачи в линии связи, связывающей конечные устройства, может варьироваться от низкоскоростной в аналоговых сетях до высокоскоростной в цифровой линии связи с различными уровнями качества обслуживания.

Протокол, в соответствии с которым принимаются параметры соединения, называется *протоколом управления линией связи* (LCP). Чтобы справиться со всеми возможными ситуациями, в протоколе PPP имеется набор стандартных параметров, действующих по умолчанию и учитывающих все стандартные конфигурации. При установлении соединения два взаимодействующих устройства пытаются сначала использовать эти параметры. Каждый конечный узел описывает свои возможности и требования. Затем на основании этой информации принимаются параметры соединения. Переговорная процедура протоколов может и не завершиться соглашением о каком-нибудь параметре. Если, например, один узел предлагает в качестве MTU (Maximum Transmission Unit — максимальный блок данных для канала) значение 1000 байт, а другой отвергает это предложение и в свою очередь предлагает значение 1500 байт, которое отвергается первым узлом, то по истечении тайм-аута переговорная процедура может закончиться безрезультатно [3].

Одним из важных параметров соединения PPP является режим аутентификации. Для целей аутентификации PPP предлагает по умолчанию протокол аутентификации по паролю (PAP), передающий пароль по линии связи в открытом виде, или протокол аутентификации по квитированию вызова (CHAP), не передающий пароль по линии связи и поэтому обеспечивающий более высокий уровень безопасности сети. Пользователям также разрешается добавлять новые алгоритмы аутентификации. Кроме того, пользователи могут влиять на выбор алгоритмов сжатия заголовка и данных.

Многопротокольная поддержка — способность протокола PPP поддерживать несколько протоколов сетевого уровня.

Внутри одного соединения PPP могут передаваться потоки данных различных сетевых протоколов, включая IP, Novell IPX и многих других.

Каждый протокол сетевого уровня конфигурируется отдельно с помощью соответствующего протокола управления сетью (NCP). Под конфигурированием понимается, во-первых, что данный протокол будет использоваться в текущем сеансе PPP, а во-вторых, переговорное согласование некоторых параметров протокола. Больше всего параметров устанавливается для протокола IP, включая IP-адреса взаимодействующих узлов, IP-адреса DNS-серверов, признак компрессии

заголовка IP-пакета и т.д. Для каждого протокола конфигурирования протокола верхнего уровня, помимо общего названия NCP, употребляется особое название, построенное путем добавления аббревиатуры CP (Control Protocol — протокол управления) к имени конфигурируемого протокола, например для IP — это протокол IPCP и т.п.

Расширяемость протокола. Под этим свойством PPP понимается как возможность включения новых протоколов в стек PPP, так и возможность применения собственных протоколов пользователей вместо рекомендуемых в PPP по умолчанию [6]. Это позволяет наилучшим образом настроить PPP для каждой конкретной ситуации.

Одной из привлекательных способностей протокола PPP является способность использования нескольких физических линий связи для образования одного логического канала, то есть агрегирование каналов. Эту возможность реализует *многоканальный протокол PPP* (MLPPP) [3].

Порядок установления соединения и разъединение показан на рис. 4.8.

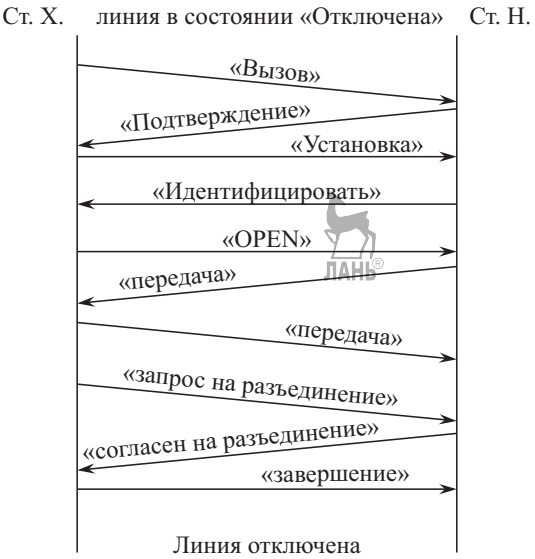


Рис. 4.8. Порядок установления соединения и разъединения

Начальное состояние протокола следующее: линия в состоянии Dead (отключена), соединения на физическом уровне нет.

После того, как физическое соединение установлено, линия переходит в состояние Establish (установка). В этот момент начинаются переговоры о параметрах с помощью протокола LCP. При успешном результате переговоров линия переходит в фазу Authenticate (иден-

тифицировать). Теперь обе стороны идентифицируют, кем является собеседник. При переходе к фазе Network (сеть) включается соответствующий протокол NCP для настройки сетевого уровня. Если настройка проходит успешно, линия переходит в фазу Open (открытая), при этом может осуществляться передача данных. Когда передача данных закончена, линия переходит к фазе Terminate (завершение), а затем снова в состояние Dead (отключена), тогда физическое соединение разрывается [7].

Таким образом, можно обобщить задачи протокола LCP (протокола управления линией связи):

- используется для переговоров о параметрах уровня передачи данных во время установочной фазы (ESTABLISH);
- определяет способ инициации процесса подачи предложения и поддержки ответных процессов принятия или отказа от поданного предложения целиком или частично;
- осуществляет проверку качества линии связи;
- позволяет отключить линию, если она больше не используется.

Формат кадра PPP для работы в ненумерованном режиме представлен на рис. 4.9.

Flag	Address	Control	Protocol	Payload	Checksum	Flag
01111110	11111111	00000011				01111110

Рис. 4.9. Формат кадра PPP для работы в ненумерованном режиме

Все PPP-кадры начинаются с флага, который составляет 1 байт и кодируется значением 01111110. Если такой байт встречается в поле данных, то применяется символьное заполнение.

Поле *Address*, которому всегда присваивается двоичное значение 11111111, которое означает, что идет групповая рассылка и все станции должны принимать этот кадр. Использование такого адреса позволяет избежать необходимости назначения адресов передачи данных.

Поле *Control* по умолчанию равно 00000011. Данное кодирование определяет ненумерованный кадр. PPP по умолчанию не обеспечивает надежной передачи с использованием порядковых номеров и подтверждений. В «зашумленных» каналах, например, при беспроводной связи, может применяться надежная передача с порядковыми номерами.

Соединение предоставляет возможность двум сторонам договориться о возможности пропускать оба поля и экономить, таким образом, по 2 байта на кадр.

Поле *Protocol* (протокол) определяет тип пакета, содержащегося в поле данных Payload (поле полезной нагрузки). Определены коды

для разных протоколов [6].

Размер поля *Protocol* по умолчанию составляет 2 байта, однако путем переговоров с помощью LCP этот размер может быть уменьшен до одного байта.

Поле данных *Payload* может быть переменной длины, вплоть до некоего оговоренного максимального значения. Если размер не оговорен во время установки соединения при помощи LCP, то по умолчанию он может составлять до 1500 байт. При необходимости данные пользователя могут дополняться специальными символами.

Поле *Checksum* (контрольная сумма), которое в обычном состоянии занимает 2 байта, но при необходимости по договоренности может занимать 4 байта.

Таким образом, PPP является механизмом формирования кадров, поддерживающим различные протоколы, которым можно пользоваться при модемных соединениях, в линиях, применяющих протокол HDLC, волоконно-оптических сетях SDH и других физических средах. PPP поддерживает обнаружение ошибок, переговоры о параметрах, сжатие заголовков, а также, по желанию, надежное соединение с использованием кадров HDLC [5, 7].

4.4. Контрольные вопросы

1. Перечислите основные задачи канального уровня, функции протоколов.
2. Назначение протокола BSC, формат кадра.
3. Назначение протокола DDCMP, формат кадра.
4. Структура кадра HDLC.
5. Принцип работы в режиме PHO, ACP.
6. Функции протокола SLIP.
7. В чем различие между байт-ориентированными и бит-ориентированными типами синхронных протоколов?
8. Укажите различия информационных, супервизорных и ненумерованных кадров.
9. В каких практических ситуациях чаще всего используется двухточечное соединение типа «точка–точка»?
10. Какими важнейшими свойствами обладает протокол PPP?
11. Опишите формат кадра байт-ориентированного протокола PPP.

4.5. Список литературы

1. *Крук Б.И., Попантонопуло В.Н., Шувалов В.П.* Телекоммуникационные системы и сети: В 3-х т.: учеб. пособие для колледжей и вузов связи. Т.1. Современные технологии / ред. В.П. Шувалов. — 4-е изд., испр. и доп. — М.: Горячая линия–Телеком, 2012. — 672 с.
2. *Будылдина Н.В., Тимченко С.В.* Системы документальной электросвязи: учебное пособие для вузов. — М.: Горячая линия–Телеком, 2011. — 198 с.
3. *Олифер В., Олифер Н.* Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. — 4-е изд. — СПб.: Питер, 2010. — 944 с.
4. [RFC 2687] Proposed Standard, PPP in a Real-time Oriented HDLC-like Framing.
5. [RFC 1144] which introduced the Van Jacobson TCP/IP Header Compression used by CSLIP.
6. [RFC 6361] PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol.

Глава 5

Протоколы сетевого и транспортного уровня

5.1. IP-протокол

Межсетевой протокол IP (Internet Protocol) — маршрутизируемый сетевой протокол семейства TCP/IP.

Протокол IP используется для доставки данных, разделяемых на пакеты, от одного узла сети к другому. Протокол не гарантирует надежной доставки пакета до адресата. В частности, пакеты могут прийти не в том порядке, в котором были отправлены, продублироваться (например, когда приходят две копии одного пакета), оказаться повреждёнными (обычно повреждённые пакеты уничтожаются) или вообще не прибыть. В этом случае, отправителю посылается соответствующее ICMP-сообщение (или не посылается ничего). Обеспечение же надежности возлагается на более высокий уровень (UDP или TCP).

На сегодняшний день на сети внедрены две версии протокола IPv4 и IPv6.

Протокол IPv4. Формат IP-пакетов показан на рис. 5.1 [1, 3].

Рис. 5.1. Формат дейтаграммы Интернет протокола версии 4

Поле *«Версия»* (4 бита) определяет версию IP-протокола (например, 4 или 6). Формат пакета определяется программой.

«Длина заголовка» (Hlen) пакета IP занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах.

Поле *полная длина* определяет полную длину IP-дейтаграммы (до 65535 октетов), включая заголовок и данные.

«Тип сервиса» TOS (Type Of Service) определяет, порядок обработки дейтаграмм. Это поле делится на 6 субполей (рис. 5.2) [1, 3].

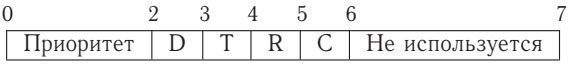


Рис. 5.2. Формат поля TOS

Субполе *«приоритет»* (3 бита) предоставляет возможность присвоить код приоритета каждой дейтаграмме.

Биты D, T, R, C характеризуют способы доставки дейтаграммы. Так, D = 1 требует минимальной задержки, T = 1 — высокую пропускную способность, R = 1 — высокую надежность, а C = 1 — низкую стоимость. Интернет не гарантирует запрашиваемый TOS, но многие маршрутизаторы учитывают эти запросы при выборе маршрута (например, протоколы OSPF и IGRP) [1].

Поле *«полная длина»* (2 байта) задает размер списка адресов, а *«указатель фрагмента»* отмечает адрес очередного маршрутизатора на пути дейтаграммы [1, 3].

Существует две формы маршрутизации: свободная маршрутизация и жесткая маршрутизация.

Жесткая маршрутизация означает, что адреса определяют точный маршрут дейтаграммы. Путь от одного адреса к другому может включать только одну сеть.

Свободная маршрутизация отличается от предшествующей возможностью пересылки между двумя адресами списка более чем через одну сеть. Маршрутизаторы имеют маршрутные таблицы, которые просматриваются каждый раз, когда маршрутизаторы получают IP дейтаграмму для отправки. Когда дейтаграмма приходит от сетевого интерфейса, по маршрутной таблице проверяется, принадлежит ли IP-адрес места назначения к списку локальных адресов или является широковещательным адресом. Если имеет место один из этих вариантов, дейтаграмма передается программному модулю в соответствии с кодом в поле протокола. IP-процессор может быть сконфигурирован как маршрутизатор, в этом случае дейтаграмма может быть переадресована в другой узел сети. Маршрутизация на IP-уровне носит пошаговый характер. Протокол IP не знает всего пути, он владеет лишь информацией о том, какому маршрутизатору послать дейтаграмму с конкретным адресом места назначения.

При просмотре маршрутной таблицы возможны варианты:

- IP-адрес назначения совпал с IP-адресами, находящимися в маршрутной таблице. В этом случае пакет будет передан соответствующему маршрутизатору или непосредственно интерфейсу адресата. Связь «точка–точка» выявляются именно на этом этапе.
- IP-адрес назначения не совпал с IP-адресами, находящимися в маршрутной таблице. В этом случае система пересылает пакет

всем узлам в сети, исключая тот, откуда пришел пакет.

- Осуществляется поиск маршрута по умолчанию и, если он найден, дейтаграмма посылается к соответствующему маршрутизатору [1, 3].

«Идентификатор», *«флаги»* и *«указатель фрагмента»*. Данные поля управляют процессом фрагментации и последующей «сборки» дейтаграммы. Идентификатор представляет собой уникальный код дейтаграммы, позволяющий идентифицировать принадлежность фрагментов и исключить ошибки при «сборке» дейтаграмм. Бит 0 поля «флаги» является резервным, бит 1 служит для управления фрагментацией пакетов (0 — фрагментация разрешена; 1 — запрещена), бит 2 определяет, является ли данный фрагмент последним (0 — последний фрагмент; 1 — следует ожидать продолжения) [3].

«Указатель фрагмента» отмечает первую свободную позицию в списке IP-адресов (куда можно произвести запись очередного адреса) [3].

«Время жизни пакетов в сети» TTL (Time To Life). Задает время жизни дейтаграммы в секундах, то есть предельно допустимое время пребывания дейтаграммы в системе. Проходя через несколько маршрутизаторов, это время уменьшается в соответствии со временем пребывания в данном устройстве или согласно протоколу обработки. Если TTL = 0, дейтаграмма из системы удаляется [1].

Поле *«Протокол»* определяет структуру поля «данные» [3].

Поле *«Контрольная сумма заголовка»* вычисляется с использованием операций сложения 16-разрядных слов заголовка по модулю 1. Сама контрольная сумма является дополнением по модулю 1 полученного результата сложения. В пакете осуществляется контрольное суммирование заголовка, а не всей дейтаграммы.

Поле *«IP-опции»* не обязательно присутствует в каждой дейтаграмме. Размер поля опции зависит от того, какие опции применены. Если используется несколько опций, они записываются подряд без каких-либо разделителей. Каждая опция содержит один байт кода опции, за которым может следовать байт длины и серия байтов данных. Если место, занятое опциями, не кратно 4 байтам, используется заполнитель.

Общие принципы адресации протокола IPv4. IP-адрес представляет собой уникальную четырехбайтовую (32-битовую) величину, выраженную в десятичных числах, разделенных точками в форме W.X.Y.Z, где точки используются для отделения байтов (например, 10.12.10.1). Поле адреса размером 32 бита состоит из двух частей: адрес сети или связи, который представляет собой сетевую часть адреса, и адрес хоста, идентифицирующий рабочую станцию в сетевом

сегменте. Разграничение сетей по количеству хостов в них традиционно осуществляется на основе так называемых классов IP-адресов. Сегодня существует 5 классов IP-адресов: A, B, C, D и E. Распределение битов в IP-адресе сети представлены в табл. 5.1 [2].

Таблица 5.1. Распределение бит в IP-адресе сети

	0				7 8				15 16				23 24				31
Класс A	1	номер сети							номер хоста								
Класс B	1	0	номер сети							номер хоста							
Класс C	1	1	0	номер сети										номер хоста			
Класс D	1	1	1	0	групповой адрес												
Класс E	1	1	1	1	0	зарезервировано											

Опираясь на эту структуру, можно подсчитать число сетей и число хостов в каждой сети (табл. 5.2) [2].

Таблица 5.2. Диапазон значения сетей

Класс	Диапазон значений первого байта	Возможное количество сетей	Возможное количество хостов в сетях
A	1 – 127	127	16 777 216
B	128 – 191	16 384	65 534
C	192 – 223	2 097 152	254
D	224 – 239	–	–
E	240 – 247	–	–

Только адреса классов A, B и C могут использоваться как уникальные. Адреса класса D применяются для обращения к набору узлов, а адреса класса E зарезервированы для исследовательских целей.

Возьмем для примера адрес в сети класса A 124.0.0.1. Здесь 124.0.0.0 представляет собой адрес сети, а единица в конце адреса обозначает первый хост в этой сети [2]. Первый и последний адреса хостов выполняют специальные функции. Так, первый адрес 124.0.0.0 (из приведенного выше примера) используется в качестве адреса сети, а последний адрес (124.255.255.255) представляет собой широковещательный адрес для этой сети. Таким образом, с помощью адресов класса A можно представить только 16 777 216 хостов в каждой сети [2].

Сети класса B определяются значениями 1 и 0 в старших битах адреса. Первые два байта в адресе (биты с 0 по 15) служат для представления адресов сетей, а оставшиеся два байта представляют номера хостов в этих сетях. В результате мы получим $2^{14} = 16\,384$ адреса сетей и 65 634 количество хостов в каждой сети (табл. 5.2) [2].

Сети класса C определяются значениями 110 в старших битах адреса. Первые три байта в адресе (биты с 0 по 23) служат для

представления адресов сетей, а оставшийся один байт представляет номера хостов в этих сетях. В результате мы получим $2^{21} = 2\,097\,150$ адреса сетей и $[(2^8 = 256) - 2]$ хостов в каждой сети (табл. 5.2) [2].

Сети класса D определяются значениями 1, 1, 1 и 0 в первых четырех битах IP-адреса. Адресное пространство класса D зарезервировано для представления групповых IP-адресов, которые используются для адресации набора узлов. Это означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса [2].

Сети класса E определяются значениями 1, 1, 1 и 1 в старших четырех битах IP-адреса. Адреса этого диапазона зарезервированы для экспериментальных целей [2].

Важным элементом адресного пространства Internet являются подсети.

Подсеть — это подмножество сети или фрагменты сети, которые не пересекаются с другими подсетями. Это означает, что сеть организации (например, сеть класса B) может быть разбита на фрагменты, каждый из которых будет составлять подсеть. Реально, каждая подсеть соответствует физической локальной сети (например, сегменту Ethernet).

Таким образом, подсети придуманы для того, чтобы обойти ограничения физических сетей на число узлов в них и максимальную длину кабеля в сегменте сети.

Например, сегмент тонкого Ethernet имеет максимальную длину 185 метров и может включать 30 узлов, а самая маленькая сеть класса C может состоять из 254 узлов. Для того чтобы достичь этой цифры, надо объединить несколько физических сегментов сети. Сделать это можно либо с помощью физических устройств (повторителей), либо при помощи машин-шлюзов.

В первом случае разбиение на подсети не требуется, так как логическая сеть выглядит как одно целое.

При использовании шлюза сеть разбивается на подсети.

Как и номера хост-машин в сетях класса A, класса B и класса C, адреса подсетей задаются локально. Обычно это выполняет сетевой администратор. Так же, как и другие IP-адреса, каждый адрес подсети является уникальным. Использование подсети никак не отражается на том, как внешний мир видит эту сеть, но в пределах организации подсети рассматриваются как дополнительные структуры.

Для примера, сеть 172.16.0.0 (рис. 5.3) разделена на 4 подсети: 172.16.1.0, 172.16.2.0, 172.16.3.0 и 172.16.4.0. Маршрутизатор определяет сеть назначения, используя адрес подсети, тем самым ограничивая объем трафика в других сегментах сети [3].

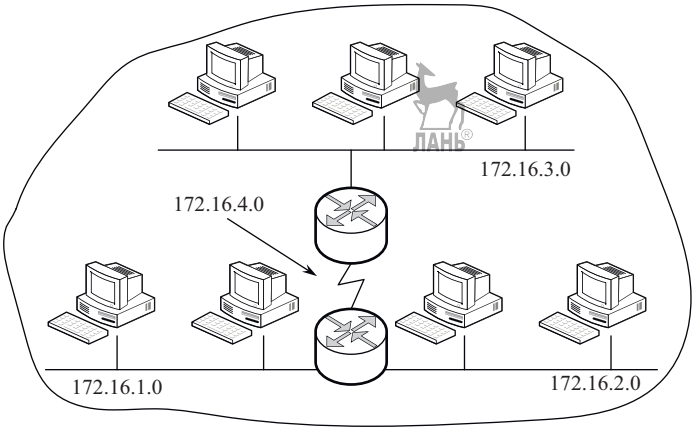
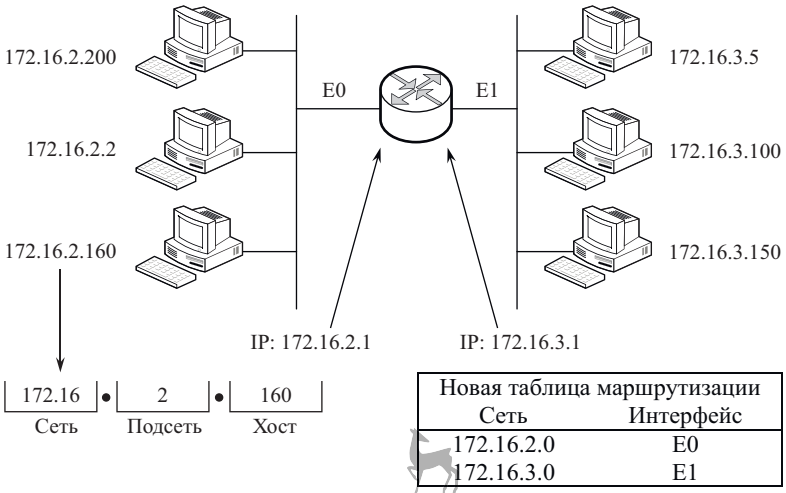


Рис. 5.3. Сеть 172.16.0.0 состоит из четырех подсетей



Новая таблица маршрутизации	
Сеть	Интерфейс
172.16.2.0	E0
172.16.3.0	E1

Рис. 5.4. Адресация подсетей расширяет сетевой номер путем создания подсетей

С точки зрения адресации, подсети являются расширением сетевого номера (рис. 5.4). Сетевые администраторы задают размеры подсетей, исходя из потребностей организации и их роста [3].

Адрес подсети включает номер сети, подсети и номер хост-машины внутри подсети. Благодаря этим трем уровням адресации подсети обеспечивают сетевым администраторам повышенную гибкость настройки.

При разбивке сетей на подсети используют ту часть IP-адреса,

которая закреплена за номерами хостов (узлов).

Таким образом, чтобы создать адрес подсети, сетевой администратор «заимствует» биты из поля хост-машин и перераспределяет их в качестве поля подсетей. Количество «заимствованных» битов можно увеличивать до тех пор, пока не останется 2 бита. Поскольку в поле хостов сетей класса B имеются только 2 октета, для создания подсетей можно заимствовать до 14 бит. Сети класса C имеют только один октет в поле хостов. Следовательно, в сетях класса C для создания подсетей можно заимствовать до 6 бит (рис. 5.5).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
11111111	11111111	111111111	11111100
Сеть	Сеть	Сеть	Поле подсети хосты

Рис. 5.5. Биты заимствуются из поля хост-машины и переопределяются в качестве поля подсети

Чем больше бит заимствуется из поля хоста, тем меньше бит в октете можно использовать для задания номера хоста. Таким образом, каждый раз, когда заимствуется 1 бит из поля хоста, число адресов хостов, которые могут быть заданы, уменьшается на степень числа 2.

Чтобы понять смысл вышесказанного, рассмотрим сеть класса C. Все 8 бит в последнем октете используются для поля хостов. Следовательно, возможное количество адресов равно 2^8 или 256.

Представим, что эту сеть разделили на подсети. Если из поля хостов заимствовать 1 бит, количество бит, которое можно использовать для адресации хостов, уменьшится до 7. Если записать все возможные комбинации нулей и единиц, можно убедиться, что число хостов, которые можно адресовать, стало равно 2^7 или 128.

Если в сети класса C из поля хостов заимствовать 2 бита, то количество бит, которое можно использовать для адресации хостов, уменьшится до 6. Общее число хостов, которое можно адресовать, станет равным 2^6 или 64.

IP-адреса, которые заканчиваются всеми двоичными единицами, зарезервированы для широковещания. Это утверждение справедливо и для подсетей.

Маскирование подсетей. Для увеличения адресного пространства в сети и создания подсетей часть IP-адреса можно замаскировать. Для этого используется «маска».

Маска подсети — это 4 байта, которые накладываются на IP-адрес для получения номера подсети; она разделена на 4 октета. Маски подсетей имеют все единицы в части, отвечающей за сети и подсети, и все нули в части, отвечающей за хост-машины. По умол-

чанию, если нет заимствованных битов, маска подсети сети класса B будет иметь вид 255.255.0.0 (рис. 5.6). Если же заимствовано 8 бит, маской подсети той же сети класса B будет 255.255.255.0. Поскольку для сетей класса B только 2 октета относится к полю хост-машин, то для создания подсетей может быть задействовано до 14 бит. В сетях класса C только один байт относится к полю хост-машин, поэтому для создания подсетей может быть заимствовано до 6 бит.

Маски подсети также используют 32-битовые IP-адреса, которые содержат все двоичные единицы в сетевой и подсетевой части адреса, и все двоичные нули в хостовой части адреса. Таким образом, адрес маски подсети класса B с 8 заимствованными битами из поля хостов будет иметь вид 255.255.255.0 (рис. 5.6). Это уже маска для сети класса C.

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
183	205	0	0
10110111	11001101	00000000	00000000
255	255	0	0
11111111	11111111	11111111	00000000
255	255	255	0

Рис. 5.6. Биты для создания подсети заимствуются из поля хост-машин, начиная со старших позиций

Для создания подсети в классе B вместо 8 бит в третьем октете заимствуются только 7. В двоичном представлении маска подсети в этом случае будет иметь вид 11111111.11111111.11111110.00000000. Следовательно, 255.255.255.0 не может больше использоваться в качестве маски подсети, в данном случае маска подсети будет 255.255.254.0.

Таким образом, стандартные маски без разбивки на подсети следующие:

класс A — 255.0.0.0;

класс B — 255.255.0.0;

класс C — 255.255.255.0.

Планирование подсетей. Сети, изображенной на рис. 5.7, присвоен адрес класса C 201.222.5.0 (первая цифра взята из диапазона первого байта класса C, вторая — любая из диапазона 1–255 свободная у провайдера, третья также из диапазона 1–255 свободная у провайдера). Предположим, необходимо организовать 20 подсетей по 5 хостов в каждой. Размер поля подсети выбирается исходя из требуемого количества подсетей. В этом примере выбор 29-битовой маски дает возможность иметь 32 подсети.

Оставшиеся биты в последнем октете используются для поля

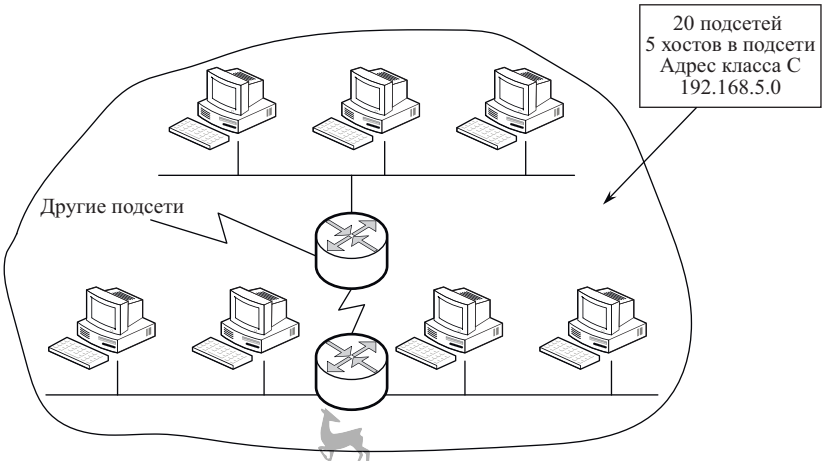


Рис. 5.7. Необходимо разделить сеть на 20 подсетей (по 5 хостов в каждой)

хост-машин. Таким образом, для определения 20 подсетей в классе С выбираем четвертый октет. Для данного примера требуемое количество хост-машин равно 5 (т.е. берем 5 бит в 4 октете, $2^5 = 32$), поэтому поле хост-машин должно содержать минимум 3 бита. Номера хост-машин могут быть 1, 2, 3 и т.д. Окончательный вид адресов формируется путем сложения начального адреса сети/подсети и номера хост-машины [5].

Таким образом, хост-машины подсети 201.222.5.16 будут адресоваться как 201.222.5.17, 201.222.5.18, 201.222.5.19 и т.д. Номер хоста 0 зарезервирован в качестве адреса кабеля, а значение номера хоста, состоящее из одних единиц, резервируется для широковещания.

Пример планирования подсетей в сетях класса В. Необходимо организовать сеть класса В. Сеть разбить на подсети. Количество подсетей 72. Определить маску, адрес 1, 2, 3 подсетей. Количество хостов в сети. Провайдер выделяет первоначальный адрес 130.10.0.0 (130 берётся из диапазона класса В, первого байта, от 128 до 191, 10 — из диапазона второго байта от 1–255). Стандартная маска класса В — 255.255.0.0.

Для определения подсетей в классе В берём третий байт. Для организации 72 подсетей необходимо взять 7 бит в третьем байте (так как $2^7 = 128$). Для определения маски подсетей берём старшие биты в третьем байте, складывая степени $2^7+2^6+2^5+2^4+2^3+2^2+2^1$, получим 254 (рис. 5.8).

Далее определим адреса 1, 2 и 3 подсетей (рис. 5.9, 5.10 и рис. 5.11). Отделяем семь бит (которые выбрали для определения 72 подсетей), переводим 1, 2 и 3 из десятичной системы в двоичную.
1 — 0000001, 2 — 0000010, 3 — 0000011 (рис. 5.9–5.11).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
11111111	11111111	11111110	00000000
255	255	254	0

Рис. 5.8. Определение маски подсети класса B для 72 подсетей

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
11111111	11111111	1111111|0	00000000
10000010	00001010	0000001|0	00000000
130	10	2	0

Рис. 5.9. Определение адреса первой подсети

Итак, адрес первой подсети будет 130.10.2.0.

Определим адрес второй подсети (рис. 5.10).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
11111111	11111111	1111111|0	00000000
10000010	00001010	0000010|0	00000000
130	10	4	0

Рис. 5.10. Определение адреса второй подсети

Адрес второй подсети будет 130.10.4.0.

Далее определим IP-адрес третьей подсети (рис. 5.11).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
11111111	11111111	1111111|0	00000000
10000010	00001010	0000011|0	00000000
130	10	6	0

Рис. 5.11. Определение адреса третьей подсети

Адрес третьей подсети будет 130.10.6.0.

В данной задаче для организации сети и подсетей было выбрано 23 бита. Адреса можно прописать следующим образом.

Адрес первой подсети будет 130.10.2.0/23.

Адрес второй подсети будет 130.10.4.0/23.

Адрес третьей подсети будет 130.10.6.0/23.

Сейчас нам необходимо определить количество хостов во всей сети.

Для организации 72 подсетей было выбрано 7 бит в третьем байте. Для определения хостов в одной подсети осталось 8 бит в четвертом байте и плюс один бит в третьем байте, итого 9 бит ($2^9 = 512$ хостов). Всего $128 \times 512 = 65\,536$ хостов может быть организовано в сети.

Необходимо вычесть из общего количества хостов два адреса:

– адрес, где присутствуют в адресе хостов все единицы, предназначены для широковещательной рассылки;
– адрес, где все нули — неопределённое значение.

Поэтому количество хостов во всей сети будет 65 534.

Рассмотрим пример определения маски подсети, адреса подсети и номера хоста в подсети и количество хостов в сети, если есть IP-адрес 120.75.83.56/21.

Определяем по первому байту, что это класс A (первый байт находится в диапазоне 1–127).

На сети и подсети отводится 21 бит, при этом маска будет 255.255.248.0 (рис. 5.12).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
11111111	11111111	11111|000	00000000
255	255	248	0

Рис. 5.12. Определение маски подсети

Далее переведём IP-адрес 120.75.83.56 из десятичной системы счисления в двоичную (рис. 5.13).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
01111000	01001011	01010011	00111000
120	75	83	56

Рис. 5.13. Перевод IP-адреса из десятичной системы в двоичную

Накладываем на IP-адрес маску (рис. 5.14).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
01111000	01001011	01010|011	00111000
120	75	83	56
11111111	11111111	11111|000	00000000
255	255	248	0

Рис. 5.14. Наложение на IP-адрес маски

Определим номер подсети и номер хоста в подсети, для этого ставим стенку (|), где заканчивается маска, т.е. отделяем 21 бит. С левой стороны увеличиваем степени от стенки, а с правой стороны, наоборот, к стенке. Так как данный адрес класса A, то первый байт не рассматриваем. Показатели степеней складываем (рис. 5.15).

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
–	01001011	01010|011	00111000
11111111	11111111	11111|000	00000000

Рис. 5.15. Определение адреса подсети и хоста в данной подсети

Адрес подсети —

$2^{11}+2^9+2^6+2^5+2^3+2^1=2048+512+64+32+8+2=2666.$

Адрес хоста —

$$2^8+2^7+2^5+2^4+2^3=256+128+32+16+8=440.$$

Определим количество хостов в сети.

Всего для подсетей выделено 13 бит (это 8 бит второго байта и 5 бит третьего байта), на хосты осталось 11 бит (это 3 бита третьего байта и 8 бит четвёртого байта):

$2^{13}+2^{11}=8192+2048=10\,240-2=10\,238$ хостов в сети.

Рассмотрим на примерах организацию подсетей.

Пример 1. В сети класса А необходимо организовать 72 подсети, определить маску подсети, адрес всей сети, который может выделить провайдер, адреса 7, 10 и 12 подсетей. Определить количество хостов в одной подсети.

Диапазон значений первого байта класса А составляет 1–127.

Стандартная маска подсети класса А: 255.0.0.0. Порядок действий следующий.

1. Для организации 72 подсетей и определения маски необходимо взять 7 бит во втором байте. 7 бит выбираем потому, что если взять 6 бит во втором байте: $2^6=64$, для создания 72 подсетей этого недостаточно. Из 128 берем 72 значения для организации необходимых нам подсетей, остальные значения адреса подсетей можно использовать для развития сети. Учитывая вышесказанное, можно определить маску сети. Маску выбираем по следующему правилу: первый байт маски класса А: 255, во втором байте выбираем 7 старших бит (рис. 5.16):

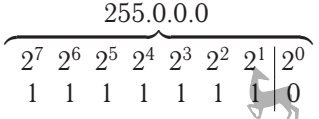


Рис. 5.16. Определение маски подсети

Складываем степени $2^7+2^6+2^5+2^4+2^3+2^2+2^1=254$.

Поэтому маска подсети будет выглядеть следующим образом: 255.254.0.0.

2. Для определения адреса сети и адресов подсетей из диапазона класса А: 1–127 выбираем любое число, например, 10 (провайдер выдает любую десятичную цифру из данного диапазона, которая у него свободна), тогда адрес сети будет 10.0.0.0, а маска, как мы уже определили ранее, 255.254.0.0 (рис. 5.17).

Вертикальная черта на рис. 5.17 ограничивает число необходимых бит, которые мы выбрали для подсетей.

10.0.0.0							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	1	1	1	0

Рис. 5.17. Определение адреса сети

Чтобы определить адреса подсетей, переведем номер каждой подсети в двоичную систему:

7 подсеть — 0111,

10 подсеть — 1010,

12 подсеть — 1100.

Полученные значения подставим к ограничительной черте и, складывая степени, определим адрес каждой подсети (рис. 5.18):

10.0.0.0							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0

Рис. 5.18. Определение адресов подсетей

7 подсеть — $2^3+2^2+2^1=14$, адрес подсети 10.14.0.0/15.

10 подсеть — $2^4+2^2=20$, адрес подсети 10.20.0.0/15.

12 подсеть — $2^4+2^3=24$, адрес подсети 10.24.0.0/15.

В полученных адресах значение /15 обозначает число занятых бит, в нашем случае восемь бит первого октета и семь бит второго октета.

3. Количество хостов во всей сети определим по следующей формуле: $(2^7+2^{17})-2=131\,198$, где степень 7 — это число использованных бит второго октета, а 17 — число свободных бит второго, третьего и четвертого октетов. Вычитаем два бита из общего числа, т.к. все нули — это неопределенное значение, а все единицы — это широковещательная рассылка.

Пример 2. Дан адрес 140.75.93.46/22. Определить класс сети, номер подсети и номер хоста в подсети. Порядок действий:

1. Для определения класса подсети смотрим на первую цифру первого байта. Число попадает в диапазон значений от 128–191 (см. табл. 5.2), а это диапазон класса B.

2. Определим номер подсети. Сеть класса B имеет стандартную маску 255.255.0.0. Учитывая, что на подсети выделено 22 бита, то маска сети с разбивкой на подсети будет определяться так: 8 бит первого байта + 8 бит второго байта + 6 бит третьего байта = 22. Следовательно, маска сети будет 255.255.252.0 (рис. 5.19).

255.255.0.0

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	1	1	0	0

Рис. 5.19. Определение маски подсети

Так как адрес сети 140.75.93.46/22 и мы уже определили, что это сеть класса B, таким образом, первые два байта — это адрес сети из табл. 5.1, и провайдер выделил два байта адреса — это 140 и 75, их переводить в двоичную систему нет необходимости. Следовательно, мы переводим из десятичной системы в двоичную цифры 93 и 46 адреса 140.75.93.46 (рис. 5.20):

93									**46**							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	1	1	0	1	.	0	0	1	0	1	1	1	0

Рис. 5.20. Перевод из десятичной системы в двоичную

Накладываем маску, которая составляет 255.255.252.0 или 22 бита (рис. 5.21):

93									**46**							
5	4	3	2	1	0	9	8									
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	1	1	0		.	0	0	1	0	1	1	1	0

Рис. 5.21. Наложение маски на адрес сети

Отделяем в 3 байте 6 бит. От ограничительной черты в третьем байте начинаем увеличивать слева степень с нуля и определяем номер подсети:

$$2^4+2^2+2^1+2^0=16+4+2+1=23.$$

С правой стороны, наоборот, увеличиваем степень к ограничительной черте и определяем номер рабочей станции:

$$2^8+2^5+2^3+2^2+2^1=302.$$

Ответ: получается 23 подсеть и в 23 подсети 302 рабочая станция.

Методы увеличения адресного пространства. Для увеличения адресного пространства применяют:

1) использование масок подсетей переменной длины;
2) введение бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR), впервые о которой было официально объявлено в 1993 году, когда были опубликованы рекомендации RFC 1517, RFC 1518, RFC 1519 и RFC 1520;
3) использование частных IP-адресов NAT;
4) внедрение протокола IPv6.

При бесклассовой адресации отсутствует привязка к классу сети, маске подсети фиксированной длины по умолчанию.

Поэтому разбиение подсетей с использованием маски переменной длины на основе технологии Variable Length Subnet Masking (VLSM) позволяет сетевому администратору разбить адресное пространство IP сети на подсети неравных размеров, в отличие от простого разбиения.

Суть технологии CIDR заключается в следующем. Каждому поставщику услуг Internet должен назначаться непрерывный диапазон в пространстве IP-адресов. При таком подходе адреса всех сетей каждого поставщика услуг имеют общую старшую часть — *префикс*, поэтому маршрутизация на магистралях Internet может осуществляться на основе префиксов, а не полных адресов сетей. Агрегирование адресов позволит уменьшить объем таблиц в маршрутизаторах всех уровней, следовательно, ускорить работу маршрутизаторов и повысить пропускную способность Internet.

При бесклассовой маршрутизации CIDR маска подсети прописывается в виде IP-адреса и через / указывается количество бит, отводимых на маску, например, 192.169.1.10/29. Число 29 соответствует маске 255.255.255.248.

5.2. Протокол IPv6

IPv6 представляет собой новую версию протокола Интернет, являющуюся преемницей версии 4. Изменения IPv6 по отношению к IPv4 можно поделить на следующие группы:

1. *Расширение адресации*. В IPv6 длина адреса расширена до 128 бит (против 32 в IPv4), что позволяет обеспечить больше уровней иерархии адресации, увеличить число адресуемых узлов, упростить авто-конфигурацию. Для расширения возможности мультикастинг-маршрутизации в адресное поле введено субполе «scope» (группа адресов). Определен новый тип адреса «anycast address» (эникастный), который используется для посылки запросов клиента любой группе серверов. Эникаст-адресация предназначена для использования с набором взаимодействующих серверов, чьи адреса не известны клиенту заранее [3].

2. *Спецификация формата заголовков*. Некоторые поля заголовка IPv4 отбрасываются или делаются опционными, уменьшая издержки, связанные с обработкой заголовков пакетов, с тем, чтобы уменьшить влияние расширения длины адресов в IPv6.

3. *Улучшенная поддержка расширений и опций*. Изменение кодирования опций IP-заголовков позволяет облегчить переадресацию пакетов, ослабляет ограничения на длину опций и делает более доступным введение дополнительных опций в будущем.

4. *Возможность пометки потоков данных*. Введена возможность помечать пакеты, принадлежащие определенным транспортным потокам, для которых отправитель запросил определенную процедуру обработки, например, нестандартный тип TOS (вид услуг) или обработка данных в реальном масштабе времени.

5. *Идентификация и защита частных обменов*. В IPv6 введена спецификация идентификации сетевых объектов или субъектов для обеспечения целостности данных и при желании защиты частной информации.

Ниже на рис. 5.22 изображен формат заголовка IPv6.

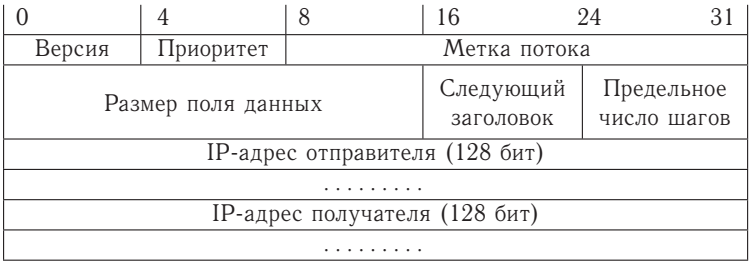


Рис. 5.22. Формат заголовка пакета IPv6

Поле *«Версия»* — 4-битный код номера версии Интернет протокола (версия Интернет протокола для IPv6 = 6).

Поле *«Приоритет»* — 4-битовое поле приоритета в IPv6 заголовке позволяет отправителю идентифицировать относительный приоритет доставки пакетов. Значения приоритетов делятся на два диапазона. Коды от 0 до 7 используются для задания приоритета трафика, для которого отправитель осуществляет контроль перегрузки (например, снижает поток TCP в ответ на сигнал перегрузки). Значения с 8 до 15 используются для определения приоритета трафика, для которого не производится снижения потока в ответ на сигнал перегрузки, например, в случае пакетов «реального времени», посылаемых с постоянной частотой.

Например, для передачи мультимедийных сообщений, где управление скоростью передачи невозможно, уровень приоритета должен лежать в пределах 8–15. Практически уровни приоритета, выше или равные 8, зарезервированы для передачи данных в реальном масштабе времени.

В случае, когда при передаче сообщений возникают перегрузки и контроль затруднен, используется нижнее значение приоритета, (8) должно использоваться для тех пакетов, которые отправитель разрешает выбросить в случае перегрузки (например, видео-трафик высокого качества), а высшее значение (15) следует использовать для

пакетов, которые отправитель не хотел бы потерять (например, аудиотрафик) [3].

«Метка потока» — 24-битный код метки потока (для мультимедиа). 24-битовое поле *метки потока* в заголовке IPv6 может использоваться отправителем для выделения пакетов, например, для сервиса реального времени (real-time). Для ЭВМ или маршрутизаторов, которые не поддерживают функцию пометки потоков, это поле должно быть обнулено при формировании пакета, сохраняться без изменения при переадресации и игнорироваться при получении.

Допускается несколько потоков между отправителем и получателем, а также обмен, не ассоциированный ни с одним из потоков. Поток однозначно описывается комбинацией адреса отправителя и ненулевой меткой потока. Пакеты, не принадлежащие ни одному из потоков, имеют метку, равную нулю.

Все пакеты, принадлежащие одному потоку, должны быть посланы одним отправителем, иметь один и тот же адрес места назначения, приоритет и метку потока.

«Размер поля данных» — 16-битовое число; оно несет в себе код длины поля данных в октетах, которое следует сразу после заголовка пакета.

«Следующий заголовок» — 8-битовый разделитель; он идентифицирует тип заголовка, который следует непосредственно за IPv6-заголовком. Использует те же значения, что и протокол IPv4.

«Предельное число шагов» (максимальное время жизни пакета в сети) — 8-битовое целое число. Уменьшается на 1 в каждом узле, через который проходит пакет. При предельном числе шагов, равном нулю, пакет удаляется.

В отличие от IPv4, узлы IPv6 не требуют установки максимального времени жизни пакетов. По этой причине поле IPv4 «timetolive» (TTL) переименовано в «hop limit» (предельное число шагов) для IPv6.

«Адрес отправителя» — 128-битовый адрес отправителя пакета.

«Адрес получателя» — 128-битовый адрес получателя пакета (возможно, не конечный получатель, если присутствует маршрутный заголовок).

Адресация IPv6. Существует три типа адресов:

Unicast: Идентификатор одиночного интерфейса. Пакет, посланный по уникастному адресу, доставляется интерфейсу, указанному в адресе.

Anycast: Идентификатор набора интерфейсов, принадлежащих разным узлам. Пакет, посланный по эникастному адресу, доставляется одному из интерфейсов, указанному в адресе (ближайшему марш-

рутизатору, по пути следования пакета в соответствии с протоколом маршрутизации). Адрес:

- выделяется из пространства unicast-адресов, существующих в данной зоне;
- определяется как отдельная маршрутная единица, т.е. должен быть уникален по всей сети;
- не может быть присвоен оконечному узлу (допустим, персональному устройству), поддерживается только маршрутизаторами;
- не может быть адресом отправителя.

Multicast: Идентификатор набора интерфейсов, принадлежащих разным узлам. Пакет, посланный по мультикастинг-адресу, доставляется всем интерфейсам, заданным этим адресом. Групповой адрес, обязательно имеет префикс /8, начинается с единиц: 11111111 (пакет доставляется всем из группы).

В IPv6 не существует широковещательных адресов, их функции переданы мультикастинг-адресам.

В IPv6 все нули и все единицы являются допустимыми кодами для любых полей, если не оговорено исключение.

IPv6 адреса всех типов ассоциируются с интерфейсами, а не узлами. Так как каждый интерфейс принадлежит только одному узлу, уникастный адрес интерфейса может идентифицировать узел.

Уникастный адрес соотносится только с одним интерфейсом. Одному интерфейсу могут соответствовать много адресов различного типа (уникастные, эникастные и мультикстные). Существует два исключения из этого правила:

1. Одиночный адрес может приписываться нескольким физическим интерфейсам, если приложение рассматривают эти несколько интерфейсов как единое целое при представлении его на уровне Интернет.

2. Маршрутизаторы могут иметь при соединении «точка–точка» ненумерованные интерфейсы (например, интерфейсу не присваивается никакого IPv6 адреса), для того чтобы исключить необходимость вручную конфигурировать и объявлять (advertise) эти адреса. Адреса не нужны для соединений «точка–точка» маршрутизаторов, если эти интерфейсы не используются в качестве точки отправления или назначения при посылке IPv6 дейтаграмм [3].

Представление записи адресов. Существует три стандартные формы для представления IPv6 адресов в виде текстовых строк.

Примеры записи:

Структура: 8 блоков по 4 шестнадцатиричных символа, например, основная форма имеет вид:

2001:0db8:0000:3456:0000:0000:defg:0987.

Из-за метода записи некоторых типов IPv6 адресов они часто содержат длинные последовательности нулевых бит. Для того, чтобы сделать запись адресов, содержащих нулевые биты, более удобной, предусмотрен специальный синтаксис для удаления лишних нулей. Использование записи «::» указывает на наличие групп из 16 нулевых бит. Комбинация «::» может появляться только при записи адреса. Последовательность «::» может также использоваться для удаления из записи начальных или завершающих нулей в адресе. Например:

Адрес уникаст 2001:db8:0:3456:0:0:defg:0987 (unicast), может быть представлен в виде:

2001:db8:0:3456::defg:0987

Адрес мультикаст FF02:0:0:0:0:0:0:1 (multicast), может быть представлен в виде:

FF02::1

Адрес обратной связи 0:0:0:0:0:0:0:1 (loopback или адрес замыкания — аналог 127.0.0.1), может быть представлено в виде:

::1

Не специфицированный адрес (адрес по умолчанию, аналог 0.0.0.0):

0:0:0:0:0:0:0:0

может быть представлен в виде:

::

Альтернативной формой записи, которая более удобна при работе с IPv4 и IPv6, является x:x:x:x:x:x:d.d.d.d, где «x» — шестнадцатеричные 16-битовые коды адреса, а «d» — десятичные 8-битовые, составляющие младшую часть адреса (стандартное IPv4-представление).

Например:

0:0:0:0:0:0:13.1.68.3

0:0:0:0:0:FFFF:129.144.52.38 или в сжатом виде [3]:

::13.1.68.3 ::FFFF:129.144.52.38.

Использование префиксов. Префикс — идентификатор оператора, сети, связи.

Например, по умолчанию префикс /64 (первые 64 байта отдаются на идентификацию провайдера).

2001:db8:0:3456::/64.

Для корпоративных и домашних провайдеров можно выделять сети с префиксом /48:

2001:247:10:1000::/64 — сеть A;

2001:247:10:2000::/64 — сеть B.

Их провайдер 2001:247:10::/48.

Типы Unicast-адресов. Unicast-адрес узла присваивается как оконечным, так и транзитным узлам (например, маршрутизаторам).

Различают:

- глобальный адрес устройства — задает уникальный адрес устройства из идентификатора сети провайдера и идентификатора интерфейса IID. Позволяет избежать использования MAC-адресов. Префикс 2000::/3;
- уникальный локальный адрес устройства (ULA, unique local address). Маршрутизируется в пределах локальной сети, но является уникальным в глобальном смысле. Префикс FC00::/7;
- Link-local — позволяет связываться устройствам одной сети напрямую. Автоконфигурируемый: префикс + IID (Interface Identifier). Префикс FE80::/10;
- адрес IPv4 перевод из IPv4 в IPv6.

Пример: 151.93.12.74 переводится в 0:0:0:0:0:FFFF:151.93.12.74 или ::FFFF:151.93.12.74

Идентификатор интерфейса IID:

- используется для формирования глобального уникального адреса;
- используется в последних 64 битах адреса (unicast) для идентификации пользователя;
- составляется из MAC-адреса устройства.

Пример: сопоставление MAC-адреса Ethernet и адреса IPv6 (см. рис. 5.23).

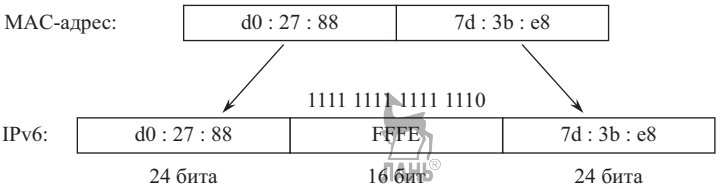


Рис. 5.23. Сопоставление MAC-адреса Ethernet и адреса IPv6

Multicast-адреса. Предназначены для рассылки всем членам группы (например, при вещании или IPTV). Не могут быть адресом отправителя.

Префикс FF00::/8. Всегда начинается с 11111111 — означает, что это групповой адрес. Структура адреса multicast-адреса (поля приведены в битах на рис. 5.24).

«Идентификатор группы»: присваивается всем узлам, входящим в группу. При этом каждый из узлов может иметь и свой уникальный адрес, иногда несколько — по количеству интерфейсов. При этом

данный адрес действителен только в области охвата.

Поле «флаги» имеет формат: XRPT, где X — резерв, R = 1 — используется как широковещательный, P = 1 — образован из IID, T = 0 — назначен для постоянного использования, T = 1 — назначен временно.

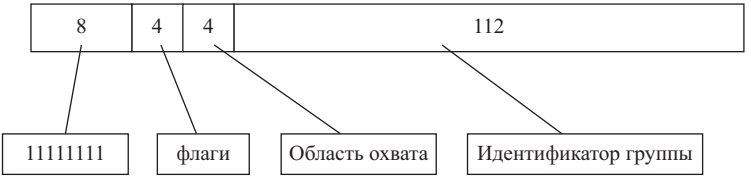


Рис. 5.24. Структура адреса multicast-адреса

Поле «область охвата» (scope) определяет применимость адреса:

- 1 — node-local (в пределах узла);
- 2 — link-local (в пределах соединения);
- 5 — site-local (в локальной сети);
- 8 — organization-local (для организации, аналог корпоративной или домовой сети);
- E — global (в общедоступной сети).

Такие адреса полностью заменяют широковещательные — понятие «широковещательный адрес» в IPv6 отсутствует. Адреса в диапазоне FF00–FF0F зарезервированы и не присваиваются.

5.3. Протокол маршрутизации RIP

Протокол RIP (Routing Information Protocol, RIP) является дистанционно-векторным протоколом внутренней маршрутизации. Протокол RIP принадлежит к классу так называемых IGP протоколов (Interior Gateway Protocol) и используется, как правило, внутри Автономной системы (Autonomous System, AS). Под AS понимается совокупность устройств, принимающих участие в работе одного протокола и находящихся под единым административным управлением. Как следует из определения, в пределах AS используется, как правило, один тип протокола маршрутизации. Примером AS может быть сеть одной организации, использующей несколько маршрутизаторов для связи офисов. Внутри такой сети может использоваться RIP (см. рис. 5.25).

RIP разрабатывался для работы в IP-сетях, объединяемых с помощью активных сетевых устройств, определяемых как маршрутизаторы. IP-сети могут использовать различные сетевые технологии канального уровня, такие как Ethernet, Token Ring, Frame Relay, линии связи «точка–точка» (PPP) и так далее [1].

Задача протокола RIP состоит в том, чтобы вовремя обновлять

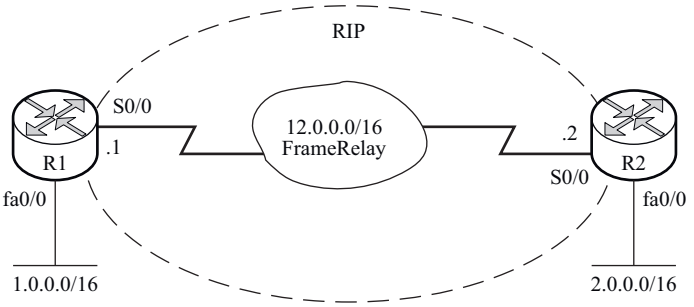


Рис. 5.25. Схема организации, работающая по протоколу маршрутизации RIP

все таблицы маршрутов в сети, посылая регулярные сообщения об их обновлении.

Процесс создания таблиц маршрутизации состоит из нескольких этапов:

- создание маршрутных таблиц;
- рассылка маршрутных таблиц соседям;
- получение RIP-сообщений от соседей и обработка информации;
- рассылка новой, уже не минимальной таблицы соседям;
- получение RIP-сообщений от соседей и обработка информации.

Если связь с какой-либо сетью обрывается, то маршрутизатор отмечает этот факт тем, что присваивает элементу вектора, соответствующему расстоянию до этой сети, максимально возможное значение, которое имеет специальный смысл — «связи нет».

Таблица маршрутизации, полученная по протоколу RIP, имеет время жизни (TTL). При поступлении очередного сообщения таймер устанавливается в исходное состояние, а затем из него каждую секунду выдается единица. Если за время тайм-аута не придет новое маршрутное сообщение об этом маршруте, то он помечается как недействительный.

В данном протоколе все сети имеют номера (способ образования номера зависит от используемого в сети протокола сетевого уровня), а все маршрутизаторы — идентификаторы. Протокол RIP широко использует понятие «вектор расстояний». Вектор расстояний представляет собой набор пар чисел, являющихся номерами сетей и расстояниями до них в хопах.

Вектора расстояний итерационно распространяются маршрутизаторами по сети, и через несколько шагов каждый маршрутизатор имеет данные о достижимых для него сетях и о расстояниях до них.

Протокол применяет широковещательные User Datagram Protocol (UDP) пакеты данных для обмена маршрутной информацией. Программное обеспечение, например Cisco IOS, посылает маршрутные обновления каждые 30 секунд, это называется рассылкой (advertising). Если маршрутизатор не получает обновления от другого маршрутизатора в течение 180 секунд или более, он помечает маршруты, обслуживаемые необновляемым маршрутизатором, как непригодные. Если через 240 секунд нет обновления, маршрутизатор удаляет все записи в таблице маршрутизации для не обновляемого маршрутизатора.

Шестикратный запас времени нужен для уверенности в том, что сеть действительно стала недоступна, а не просто произошли потери RIP-сообщений.

Работа протокола основана на алгоритме Беллмана–Форда (Bellman–Ford algorithm), или distance-vector алгоритме (distance-vector — вектор-дистанция)

Пример работы сети на основе протокола RIP рассмотрим на рис. 5.26, где последовательно соединены четыре маршрутизатора. Сеть 1 непосредственно присоединена к маршрутизатору A.

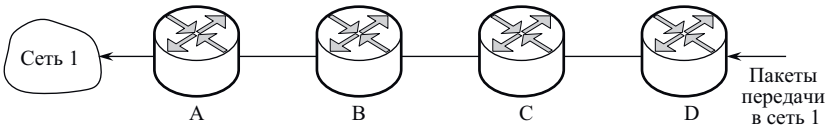


Рис. 5.26. Сеть из последовательно соединенных маршрутизаторов

Согласно алгоритму Беллмана–Форда, маршрутизатор B получает информацию о пути в Сеть 1 от маршрутизатора A, увеличивает метрику (шаг) до единицы. Добавляет 1 к значению вектора расстояния и затем посылает копию таблицы маршрутизации маршрутизатору C. В свою очередь, маршрутизатор C повышает значение метрики до 2 и обменивается маршрутной информацией с маршрутизатором D, который увеличивает значение метрики до 3. Таким образом, результирующий вектор (количество шагов или хопов) или расстояние в сети поэтапно увеличивается.

Данная особенность работы алгоритма может приводить к появлению маршрутных петель в случае медленной сходимости после изменений в сети. Предположим, что до изменений наилучшим путем к Сети 1 для маршрутизатора D был путь через маршрутизаторы C и B (см. рис. 5.27). Метрика пути из маршрутизатора D в Сеть 1 была равна 3 шагам (через маршрутизаторы C и B). Если вышла из строя, например, Сеть 1 (рис. 5.28), то начинается обновление маршрутной информации. При этом может возникнуть маршрутная петля в связи с тем, что:

1. Маршрутизатор A посылает обновление об изменении маршрутов маршрутизатору B и он прекращает передачу пакетов данных в Сеть 1. Но так как маршрутизаторы C, E и D еще не получили обновления, они продолжают передачу.

2. Маршрутизатор B отправляет обновления маршрутизаторам C и E, они прекращают отправлять пакеты в Сеть 1, но маршрутизатор D продолжает передачу. Он пока считает, что данный путь до Сети 1 существует через маршрутизатор C и метрика равна 3 переходам.

3. Возможно, что маршрутизатор D отправит обновление маршрутизатору E, в нем он укажет, что существует маршрут в Сеть 1 через маршрутизатор C, но метрика при этом будет равна 4 переходам (от E через D, C, B и A).

4. Исходя из этого, маршрутизатор E обновит свою таблицу маршрутизации и перешлет обновление маршрутизатору B с метрикой в 5 переходов, и так далее по кольцу.

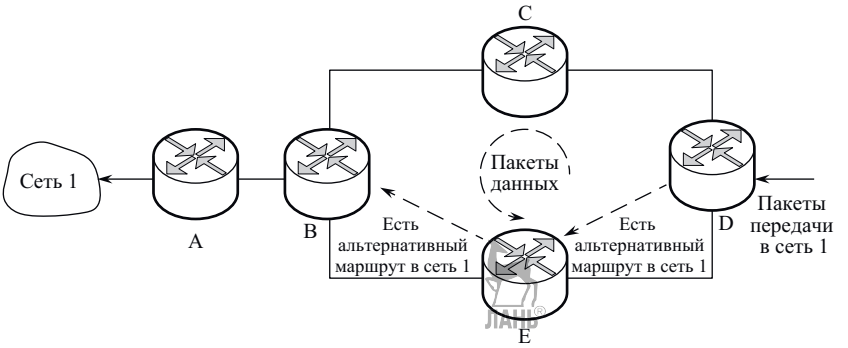


Рис. 5.27. Образование маршрутных петель в сети

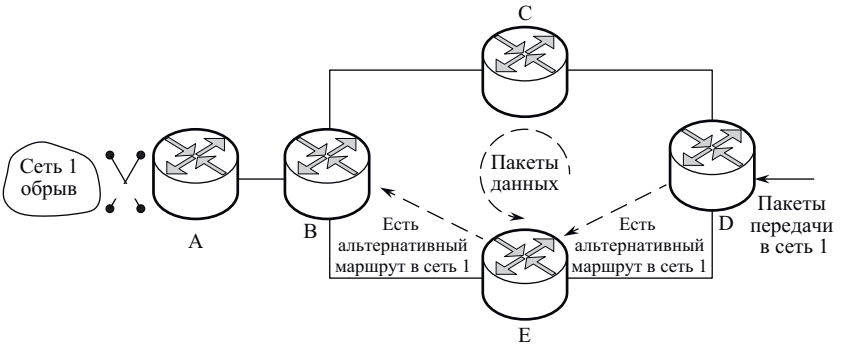


Рис. 5.28. Образование маршрутных петель в сети, если, например, вышла из строя Сеть 1

5. В этом случае любой пакет, предназначенный Сети 1, будет передаваться по кольцу (по петле) от маршрутизатора D к маршрутизатору C, затем к B, E и снова D.

Таким образом, образовалась маршрутная петля, из которой пакет не может выйти, если не принять специальных мер. А так как сведения о недоступных через отказавший маршрутизатор сетях распространяются по сети не очень быстро, время распространения кратно времени жизни записи, при этом коэффициент кратности равен количеству шагов между самыми дальними маршрутизаторами сети. Поэтому одна из причин выбора в качестве периода рассылки широковещательной копии маршрутной таблицы небольшой величины всего 30 секунд. Т.е. раз в 30 секунд каждый маршрутизатор посылает широковещательную копию своей маршрутной таблицы всем соседям-маршрутизаторам, с которыми непосредственно связан.

Движение по петле теоретически может быть бесконечным. Однако в существующих протоколах имеется ряд средств, чтобы предотвратить бесконечную циркуляцию пакетов по петле маршрутизации [10].

1. В протоколе RIP максимальное значение метрики не может превышать 15. Поэтому как только при обмене маршрутной информацией (рис. 5.28) возрастающая на каждом шаге метрика достигает значения 16, Сеть 1 будет считаться недостижимой и пакет отбрасывается.

2. В заголовке протокола сетевого уровня IP имеется поле TTL — время жизни пакета в сети. Это значение фиксируется каждым маршрутизатором при прохождении пакета и вычитается 1. Таким образом, число устройств, через которые может пройти пакет, ограничено. При обнулении значения TTL маршрутизатор отбрасывает пакет и отправителю с помощью протокола ICMP посылается сообщение о недостижимости сети.

3. Принцип расщепления горизонта (split horizon) также позволяет бороться с маршрутными петлями. При описании возникновения маршрутной петли (рис. 5.28) показано, что если маршрутизатор D отправит обновление маршрутизатору E и в нем укажет, что есть альтернативный маршрут в Сеть 1 через маршрутизатор C, то маршрутизатор E модернизирует свою таблицу маршрутизации и перешлет обновление маршрутизатору B. Таким образом, маршрутизатор B может ошибочно считать, что имеется путь к Сети 1, но с худшей метрикой. Однако ранее маршрутизатор B уже получил от маршрутизатора A информацию, что Сеть 1 недостижима. Принцип расщепления горизонта указывает, что нельзя посылать информацию маршрутизатору B о Сети 1 в обратном направлении, т.е. от маршрутизатора C или E.

4. Пометка недоступного маршрута запрещенной метрикой (route poisoning). В этом случае маршрутизатор, имеющий какой-то маршрут к сети, сразу же после получения сообщения о недостижимости данной сети включает в соответствующую строку таблицы маршрутизации запрещенное значение метрики, равное 16. Обычно этот метод используется совместно с принципом расщепления горизонта и механизмом мгновенной рассылки объявлений об изменении топологии сети.

5. При методе мгновенных обновлений (triggered update) их рассылка производится сразу, как только маршрутизатор обнаружит какие-либо изменения в сети, не дожидаясь окончания периода обновления. Последующие маршрутизаторы также мгновенно рассылают информацию об изменении в сети. Это приводит к ускорению сходимости сети.

6. Таймер удержания информации (holddown timer) запускается на маршрутизаторе, когда от соседнего устройства приходит информация о том, что ранее доступная сеть становится недоступной. Это дает больше времени для распространения информации об изменениях по всей сети. Возможны разные варианты действия протокола вектора расстояния:

- если до истечения времени таймера удержания информации от того же устройства приходит обновление, что сеть снова стала достижимой, то протокол помечает сеть как доступную и выключает таймер;
- если до истечения времени таймера приходит обновление от другого маршрутизатора с лучшей метрикой, чем была ранее, то протокол помечает сеть как доступную и выключает таймер;
- если до истечения времени таймера приходит обновление от другого маршрутизатора с худшей метрикой, то это обновление игнорируется.

Таким образом, указанные меры борьбы с маршрутными петлями позволяют маршрутизаторам избегать их. Но время сходимости протокола RIP велико по сравнению с протоколами состояния канала (link state). Поэтому протокол RIP используется только в малых сетях. Данный протокол имеет важное достоинство: для его функционирования требуется существенно меньший объем оперативной памяти и быстродействие центрального процессора. Вот почему данный протокол разработан для новой версии протокола IPv6 [10].

Для обеспечения маршрутизации на основе префикса CIDR (Classless Inter-Domain Routing — бесклассовая междоменная маршрутизация), выполнения аутентификации, поддержки подсетей и использования групповой передачи разработан и эксплуатируется про-

токол вектора расстояния RIPv2. Однако все другие параметры у него аналогичны протоколу RIPv1.

Протокол RIP-2, описан в документе RFC 1388. На рис. 5.29 приведен формат сообщения второй версии протокола RIP.

Команда (8 бит)	Версия (8 бит)	Домен маршрутизации (16 бит)
Идентификатор адресной схемы (16 бит)		Метка маршрута (16 бит)
IP-адрес (32 бита)		
Маска подсети (32 бита)		
Следующий переход (32 бита)		
Метрика (32 бита)		

Рис. 5.29. Формат сообщения протокола RIP-2

«Команда» (8 бит). Поле содержит число, обозначающее либо запрос, либо ответ. Команда-запрос запрашивает хост или маршрутизатор об отправке всей таблицы маршрутизации или ее части. Пункты назначения, для которых запрашивается ответ, перечисляются далее в данном пакете. Ответная команда представляет собой ответ на запрос или какую-нибудь не затребованную регулярную корректировку маршрутизации. Отвечающая система включает в ответный пакет всю таблицу маршрутизации или ее часть. Регулярные сообщения о корректировке маршрутизации включают в себя всю таблицу маршрута.

«Версия» (8 бит). Поле версии определяет реализуемую версию RIP. Поскольку в сети возможны многие реализации RIP, это поле может быть использовано для сигнализации о различных потенциально несовместимых реализациях.

Поле *«Домен маршрутизации»* (16 бит) используется вместе с полем *«Следующий переход»* для того, чтобы автономные системы могли разделить одну физическую среду передачи.

«Идентификатор адресной схемы» (16 бит). Это поле определяет конкретное семейство адресов. В сети Internet этим адресным семейством обычно является IP (значение равно 2), но могут быть также представлены другие типы сетей.

Поле *«Метка маршрута»* (16 бит) предназначено для сигнализации внешних маршрутов и используется протоколами политики маршрутизации (EGP или BGP).

Поле *«Маска подсети»* (32 бита) позволяет выполнять маршрутизацию в сформированной структуре подсетей.

Поле *«Метрика»* также занимает 4 байта в заголовке дейтаграммы протокола RIP и может принимать значения от 0 до 15 [3].

5.4. Внутренний протокол маршрутизации OSPF

Протокол маршрутизации OSPF (Open Shortest Path First) представляет собой открытый (Open) протокол состояния связей, исполь-

зующий алгоритм SPF поиска кратчайшего пути на графе. Применяется для внутренней маршрутизации в сетях любой сложности.

OSPF предлагает решение следующих задач:

- увеличение скорости сходимости;
- поддержка сетевых масок переменной длины (Variable Length Subnet Masking, VLSM,);
- достижимость сети (быстро обнаруживаются отказавшие маршрутизаторы);
- оптимальное использование пропускной способности;
- динамическое перераспределение трафика между параллельными каналами, которое выполняется пропорционально степени загруженности этих каналов;
- метод выбора пути.

Преимущество протокола OSPF:

- отсутствуют ограничения на размер сети, используется иерархическая структура сети, что позволяет существенно повысить эффективность использования каналов передачи данных за счет сокращения доли передаваемого по ним служебного трафика;
- обеспечивает несколько маршрутов в сторону одного узла;
- обеспечивает аутентификацию;
- обеспечивает поддержку внеклассовых сетей;
- обеспечивает передачу обновлений маршрутов с использованием адресов типа multicast;
- обеспечивает достаточно большую скорость установления маршрута;
- обеспечивает использование процедуры аутентификации при передаче и получении обновлений маршрутов.

Основные особенности протокола:

- каждому каналу может быть присвоен свой вес (количество ретрансляций). Ограничение на количество ретрансляций («хопов») — 65 535;
- каждый узел содержит базу сетевых путей в виде дерева, в вершине которого находится данный узел;
- если существуют пути с одинаковым весом, нагрузка распределяется между ними (режим баланса нагрузки);
- широковещательная рассылка таблиц маршрутизации производится только при появлении изменений;
- сообщения об изменениях в таблице маршрутизации отправляются только тем маршрутизаторам, которые непосредственно связаны с ним. Метод «прочти сам и передай дальше» уменьшает нагрузку на сеть.

Так как OSPF реализован в качестве единственного протокола маршрутизации в сети («однородная маршрутизирующая система»), где каждый маршрутизатор поддерживает свою собственную таблицу маршрутизации, поэтому он должен хранить информацию только о непосредственно подключенных к нему подсетях и тех маршрутизаторах, которые ему непосредственно доступны (так называемых смежных маршрутизаторах).

Процесс построения таблиц маршрутизации разбит на два этапа.

1-й этап. Каждый маршрутизатор строит граф связей сети. Для этого все маршрутизаторы обмениваются сообщениями со своими соседями — объявлениями о состоянии связей. При этом маршрутизаторы ее не модифицируют, а передают в неизменном виде. В результате все маршрутизаторы обладают идентичными сведениями о графе сети, которые хранятся в базе данных о топологии сети.

2-й этап. Нахождение оптимальных маршрутов на основе итерационного алгоритма Дейкстры. В каждом найденном маршруте запоминается один шаг — до следующего маршрутизатора, эта информация попадает в таблицу маршрутизации. Если несколько маршрутов имеют одну и ту же метрику — запоминаются первые шаги для всех этих маршрутов.

Для контроля состояния связей маршрутизаторы передают друг другу каждые 10 секунд короткие сообщения HELLO (см. рис. 5.30).

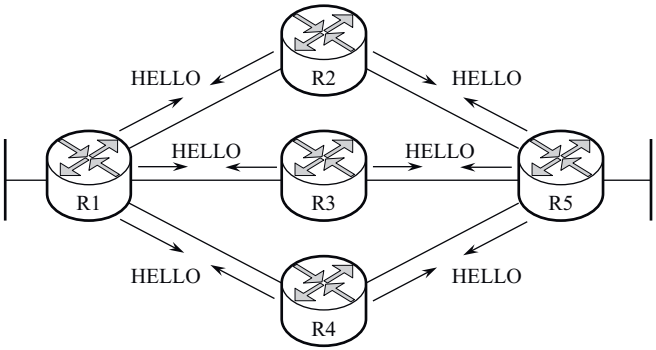


Рис. 5.30. Контроль состояния связей

Таким образом, тестируется состояние линий. Если в течение определенного периода сообщения от какого-то маршрутизатора-соседа перестают поступать — данный маршрутизатор делает вывод о неработоспособном состоянии связи, корректирует свои базы данных и шлет объявления об изменении состояния линий своим непосредственным соседям. Те тоже корректируют свои базы данных и пересылают информацию дальше.

Аналогичная процедура происходит, если появляется новый сосед и заявляет о себе сообщением HELLO.

Если состояние сети не меняется — объявления о связях не генерируются, что экономит пропускную способность сети и вычислительные ресурсы маршрутизаторов.

Каждые 30 минут все маршрутизаторы обмениваются всеми записями базы данных о топологии сети с целью синхронизации для более надежной работы.

Для сбора маршрутной информации протокол OSPF использует извещения состояния канала (LSA, link-state advertisement), топологическую базу данных, алгоритм SPF, результирующее SPF-дерево и таблицу маршрутизации путей и портов в каждую сеть [11].

Метрика представляет собой оценку качества связи в данной сети (на данном физическом канале). Метрика учитывает следующие параметры:

- пропускную способность канала;
- величину задержки распространения сигнала в канале;
- надежность канала;
- загруженность канала;
- размер максимального блока данных, который может быть передан через данный канал.

Чем меньше метрика, тем выше качество соединения. Метрика маршрута равна сумме метрик всех связей (сетей), входящих в маршрут. В простейшем случае метрика каждой сети равна единице, тогда метрика маршрута является его длиной, определяемой количеством шагов до станции назначения (т.е. количеством маршрутизаторов, через которые будет проходить путь). При использовании алгоритма SPF ситуации, приводящие к счету до бесконечности, отсутствуют. Значения метрик могут варьироваться в широком диапазоне. Протокол OSPF позволяет назначить для любой сети различные значения метрик в зависимости от типа сервиса. (Тип сервиса запрашивается дейтаграммой в соответствии со значением поля TOS ее заголовка).

Для каждого типа сервиса будет вычисляться свой маршрут, и дейтаграммы, затребовавшие наиболее скоростной канал, могут быть отправлены по одному маршруту, а затребовавшие менее скоростной канал — по-другому. Метрика пути, оценивающая пропускную способность, определяется как количество секунд, требуемое для передачи через физическую среду данной сети. Порядок расчета метрик, оценивающих надежность, задержку и стоимость, не определен. Администратор, желающий поддерживать маршрутизацию по этим типам сервисов, сам назначает разумные и согласованные метрики по этим параметрам. Если не требуется маршрутизация с учетом типа

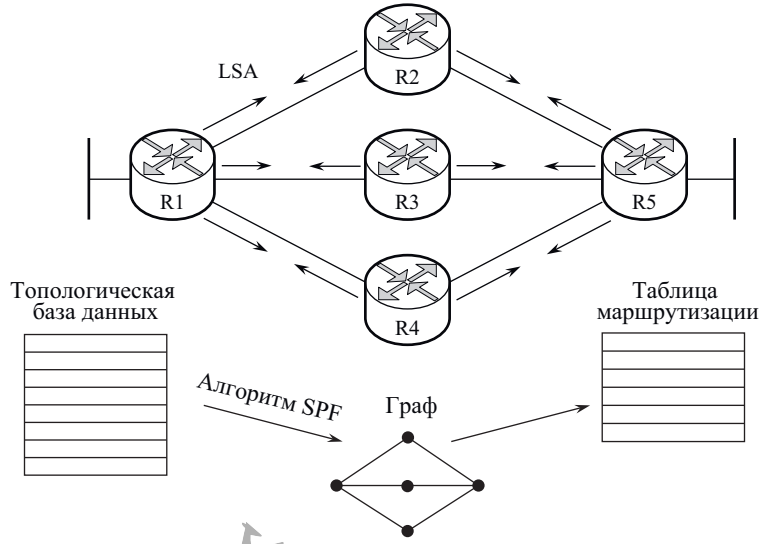


Рис. 5.31. Построение топологической базы

сервиса (или маршрутизатор ее не поддерживает), используется метрика по умолчанию, равная метрике по пропускной способности.

Дадим краткое описание работы протокола на основе алгоритма SPF. Маршрутизаторы обмениваются hello-пакетами через все интерфейсы, на которых активирован OSPF. Маршрутизаторы, разделяющие общий канал передачи данных становятся соседями, когда они приходят к договоренности об определенных параметрах, указанных в их hello-пакетах.

1. Пара маршрутизаторов, находящихся в состоянии соседства, синхронизирует между собой базу данных состояния каналов (рис. 5.31).

2. Каждый маршрутизатор, используя LSA, строит топологическую базу данных состояния каналов (топологическую таблицу), которая является картиной связи маршрутизаторов в одной области, обновляет ее и передает LSA всем соседним устройствам. Маршрутизаторы внутри одной области обладают общей информацией, у них одинаковые топологические базы данных.

Канал — это линия связи или интерфейс, соединяющий один маршрутизатор с другим или с сетью. Состояние канала — это описание интерфейса и его связей с соседними маршрутизаторами. Описание интерфейса может включать, например, IP-адрес интерфейса, маску, тип сети, к которой он подключен. Набор всех этих состояний каналов формирует базу данных состояния каналов (рис. 5.32).

3. Как только маршрутизаторы OSPF соберут информацию о со-

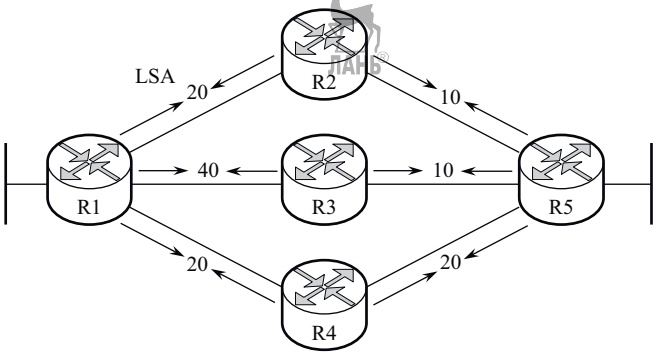


Рис. 5.32. Состояние канала и назначение метрик

стоянии каналов, они начинают вычислять кратчайший путь к каждой сети на основе метрик. Каждый маршрутизатор считает себя корнем дерева и, используя базу данных состояний каналов, вычисляет наилучшие пути к сетям назначения, применяя алгоритм SPF (алгоритм Дейкстры) и выстраивая при этом SPF-дерево, основываясь на суммарной стоимости маршрута, который используется для достижения этих сетей. Данный процесс может обнаруживать изменения в сетевой топологии, вызванные отказами оборудования и ростом сети.

Каждый маршрутизатор будет иметь свой собственный взгляд на топологию, несмотря на то, что все маршрутизаторы будут строить дерево кратчайших путей, используя одну и ту же базу данных состояния каналов (см. рис. 5.33).

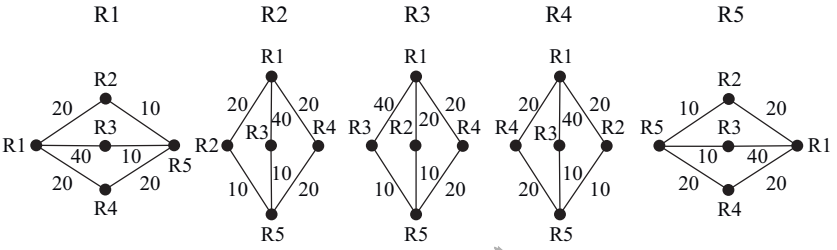


Рис. 5.33. Свой собственный взгляд на топологию со стороны маршрутизаторов

4. Затем из этого дерева к сетям назначения выбираются пути с наименьшей стоимостью (метрикой) и помещаются в таблицу маршрутизации. Метрика интерфейса — это индикатор усилий, которые необходимы для отправки пакета через этот интерфейс. Метрика интерфейса обратно пропорциональна полосе пропускания интерфейса, таким образом, бóльшая полоса пропускания соответствует меньшей метрике.

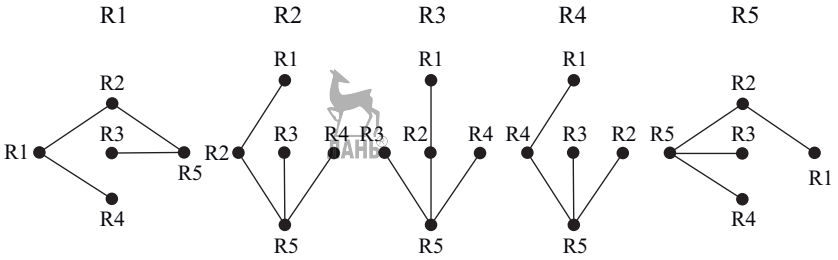


Рис. 5.34. Выбираются пути с наименьшей стоимостью

В нашем примере мы будем использовать метрики, указанные на рис. 5.32, без учета типов сервиса. Следует заметить, что маршрутизация по типам сервиса редко используется, более того, она исключена из последних версий стандарта OSPF.

Для работы алгоритма SPF на каждом маршрутизаторе, как уже говорилось ранее, создается база данных состояния связей, представляющая собой полное описание графа OSPF-системы. При этом вершинами графа являются маршрутизаторы, а ребрами — соединяющие их связи. Базы данных на всех маршрутизаторах одинаковы.

База данных состояния связей представляет из себя таблицу, где для каждой пары смежных вершин графа (маршрутизаторов) указано ребро (связь), их соединяющее, и метрика этого ребра. База данных состояния связей в нашем примере выглядит, как показано в табл. 5.3.

Таблица 5.3. База данных состояния связей

От → до	шаг	Метрика (стоимость)
R1 → R2	1	20
R1 → R3	1	40
R1 → R4	1	20
R2 → R5	1	10
R3 → R5	1	10
R4 → R5	1	20
R1 → R2 → R5	2	30
R1 → R4 → R5	2	40
R1 → R3 → R5	2	50
R1 → R2 → R5 → R3	3	40
.....	.....	.....

Алгоритм SPF, основываясь на базе данных состояния связей, вычисляет кратчайшие пути между маршрутизаторами. Результатом работы алгоритма является таблица, где для каждой вершины графа указан список ребер, соединяющих маршрутизаторы между собой по кратчайшему пути.

5. После первоначального построения таблицы маршрутизации необходимо отслеживать изменения состояния сети и вносить коррективы в таблицу маршрутизации. Для контроля состояния связей и соседних маршрутизаторов они передают специальные короткие сообщения HELLO. Если состояние сети не меняется, то маршрутизаторы корректировкой своих таблиц маршрутизации не занимаются и не посылают соседям объявления о связях. Сообщения HELLO отправляются через каждые 10 секунд, чтобы повысить скорость адаптации маршрутизаторов к изменениям, происходящим в сети. Небольшой объем этих сообщений делает возможной такое частое тестирование состояния соседей и связей с ними.

Если же состояние связи изменилось, то начинается лавинная рассылка LSA по всей сети, касающаяся только данной связи, что экономит пропускную способность сети (рис. 5.35). Получив новое объявление об изменении состояния связи, маршрутизатор пересчитывает дерево и заново ищет оптимальные маршруты. Одновременно маршрутизатор ретранслирует объявление каждому из своих ближайших соседей (кроме того, от которого он получил это объявление).

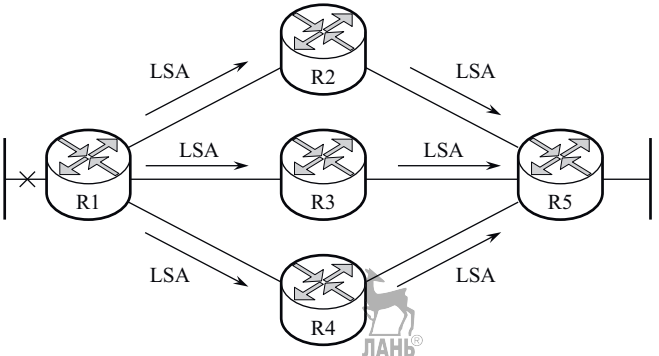


Рис. 5.35. Лавинная рассылка LSA

Поддержка множественных маршрутов (multipath). Если между двумя узлами сети существует несколько маршрутов с одинаковыми или близкими по значению метриками, протокол OSPF позволяет направить часть трафика по этим маршрутам в пропорции, соответствующей значениям метрик. Например, если существует два альтернативных маршрута с метриками 1 и 2, то две трети трафика будет направлено по первому из них, а оставшаяся треть — по второму.

Положительный эффект от использования такого механизма заключается в уменьшении средней задержки прохождения дейтаграмм между отправителем и получателем, а также в уменьшении колеба-

ний значения средней задержки. Менее очевидное преимущество поддержки множественных маршрутов состоит в следующем. Если при использовании только одного из возможных маршрутов этот маршрут внезапно выходит из строя, весь трафик будет разом перемаршрутизирован на альтернативный маршрут, при этом во время процесса массового переключения больших объемов трафика с одного маршрута на другой весьма велика вероятность образования затора на новом маршруте. Если же до аварии использовалось разделение трафика по нескольким маршрутам, отказ одного из них вызовет перемаршрутизацию лишь части трафика, что существенно сгладит нежелательные эффекты [3, 9].

OSPF-заголовок. Размер OSPF-сообщения ограничен максимальным размером дейтаграммы. Все сообщения OSPF имеют общий заголовок, следующий в дейтаграмме непосредственно за IP-заголовком (рис. 5.36).

0 7	15	23 24 31
Версия	Тип сообщения	Длина сообщения
Идентификатор маршрутизатора		
Идентификатор области		
Контрольная сумма		Аутентификация сообщения
Аутентификационные данные		

Рис. 5.36. Формат заголовка OSPF

Рассмотрим назначение полей.

Поле *«Версия»* (1 байт) — версия протокола.

Поле *«Тип сообщения»* (1 байт). Применяются следующие типы сообщений:

- hello — используется для «знакомства» с соседями;
- описание базы данных (Database Description) — сообщает о том, насколько свежей информацией располагает отправитель;
- запрос состояния связей (Link State Request) — запрашивает информацию у партнера;
- обновление состояния связей (Link State Update) — сообщает соседям информацию о связях отправителя;
- подтверждение приема сообщения о состоянии связей (Link State Acknowledgment) — подтверждает обновление состояния связей.

«Длина сообщения», включая заголовок, составляет 2 байта.

«Идентификатор маршрутизатора», отправившего сообщение (4 байта), равен адресу одного из IP-интерфейсов маршрутизатора.

«Идентификатор области», к которой относится данное сообщение (4 байта), — номер 0 зарезервирован для магистрали. Часто

идентификатор области полагают равным адресу IP-сети (одной из IP-сетей) этой области.

«Контрольная сумма» (2 байта) — охватывает все OSPF-сообщение, включая заголовок, но исключая поле «аутентификационные данные»; вычисляется по тому же алгоритму, что и в IP-заголовке.

«Аутентификации сообщения» (2 байта). Стандарт определяет несколько возможных типов, самые простые из них: 0 — нет аутентификации, 1 — аутентификация с помощью пароля.

«Аутентификационные данные» (может быть от 4 до 7 байт) — например, 8-символьный пароль.

Каждый маршрутизатор самостоятельно производит выборы выделенного и запасного выделенного маршрутизаторов на основании имеющихся у него данных о соседях и о том, кого каждый из соседей назначил на эту роль. Фактически процесс выборов происходит постоянно, после получения каждого Hello-сообщения, но алгоритм гарантирует, что при стабильном состоянии сети всеми маршрутизаторами будут выбираться одни и те же идентификаторы выделенного маршрутизатора (Designated Router, DR) и идентификаторы запасного выделенного маршрутизатора (Backup Designated Router, BDR).

Каждый маршрутизатор может объявить себя либо выделенным, либо запасным, поместив свой идентификатор в соответствующее поле своих Hello-сообщений. Иначе он может поместить туда адреса других маршрутизаторов, если он считает их занимающими соответствующие роли. Если маршрутизатор не определился с выбором DR и/или BDR (например, после включения), он заполняет соответствующие поля нулями.

Выбор проводится только среди соседей, с которыми установлена двусторонняя связь и приоритет которых не равен нулю; в этот список маршрутизатор включает и себя, если его приоритет не нулевой [3, 9].

5.5. Протокол BGP-4

Протокол граничного шлюза (BGP, Border Gateway Protocol) определенный в RFC 1771, является протоколом маршрутизации между автономными системами (AS).

Автономная система (AS, Autonomous Systems) — это одна или несколько сетей, имеющих одну политику маршрутизации.

Основной функцией протокола BGP является обмен информацией о доступности сетей с другими системами BGP. Информация о доступности сетей включает список автономных систем (AS), через которые проходит эта информация. Этих сведений достаточно для построения графа связности AS, из которого могут исключаться маршрутные петли (routing loop), а также для принятия некоторых решений на уровне политики AS [12].

Главная цель BGP — сократить транзитный трафик. Он выполняет задачи по управлению маршрутизацией в глобальных сетях, таких как Интернет. BGP — очень устойчивый и хорошо масштабируемый, проявляет исключительную стабильность в маршрутизации между автономными системами (даже при огромных таблицах маршрутизации) и предоставляет сетевым администраторам большую свободу действий и гибкость в создании правил маршрутизации.

Протокол BGP использует расширенное понятие автономной системы. В данном случае внутри автономной системы шлюзы могут использовать несколько различных протоколов маршрутизации и несколько метрик. Однако внутри автономной системы должен существовать единый план маршрутизации, позволяющий рассматривать автономную систему как единое целое (рис. 5.37).

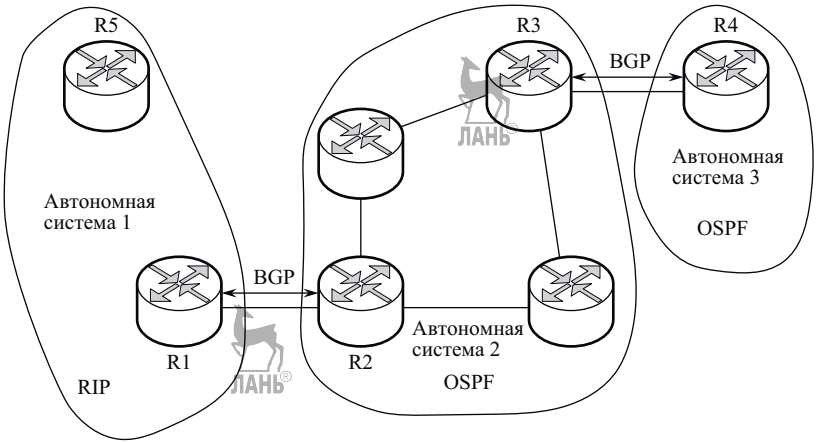


Рис. 5.37. Расширенное понятие автономной системы

В зависимости от того, с каким трафиком имеет дело автономная система, она причисляется к одной из следующих категорий:

- тупиковая автономная система, имеющая единственное соединение с другими автономными системами; фактически, данная система имеет дело только с локальным трафиком;
- многовходовая автономная система. Эта система имеет более одного соединения с другими автономными системами, но она отказывается поддерживать транзитный трафик;
- транзитная автономная система, которая имеет более одного соединения с другими автономными системами и предназначена для поддержания обоих видов трафика.

Протокол BGP использует в качестве транспортного протокола протокол TCP. Два маршрутизатора BGP создают соединение TCP

друг с другом. Эти маршрутизаторы являются равноправными. Равноправные маршрутизаторы обмениваются сообщениями для инициирования и подтверждения параметров соединения. Хост-ЭВМ, использующие протокол BGP, не обязательно должны одновременно являться шлюзами. Хост-ЭВМ, не являющаяся шлюзом, может обмениваться маршрутной информацией со шлюзами при помощи протокола EGP или внутреннего протокола маршрутизации. Данная хост-ЭВМ может затем использовать протокол BGP для обмена маршрутной информацией с граничным шлюзом другой автономной системы.

Среди используемых в настоящее время протоколов маршрутизации BGP отличается тем, что применяет информацию о векторе (направлении) и о пути к пункту назначения. С этой же целью другие протоколы маршрутизации (такие как OSPF, IS-IS) используют метрики или стоимости маршрутов в сочетании с некоторой долей информации о топологии сети.

Протокол BGP является протоколом вектора маршрута и используется для обмена маршрутной информацией между автономными системами. Термин вектор маршрута (path vector) происходит из самого принципа действия BGP: маршрутная информация содержит последовательности номеров AS, через которые прошел пакет с заданным префиксом сети. Маршрутная информация, связанная с префиксом, используется для профилактики образования петель в маршрутах.

Пример работы дистанционно-векторного протокола маршрутизации изображен на рис. 5.38.

Предположим, что маршрутизатор A сгенерировал маршрут к сети 10.1.1.0/24 и объявил его маршрутизатору B. В информации о том, как достичь сети назначения 10.1.1.0/24, маршрутизатор A указывает, что он является первым маршрутизатором в пути. Маршрутизатор B, получив этот маршрут, добавляет себя в путь и отправляет его маршрутизатору C, который, в свою очередь, добавляет себя в путь к

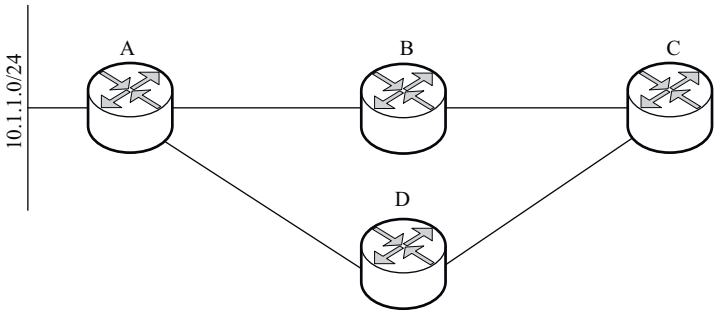


Рис. 5.38. Пример работы дистанционно-векторного протокола маршрутизации

сети 10.1.1.0/24 и отправляет маршрут маршрутизатору D.

Когда маршрутизатор D получает маршрут к пункту назначения 10.1.1.0/24, он обнаруживает, что путь к нему проходит через маршрутизаторы C, B и A. Маршрутизатор D добавляет себя в путь и отсылает полученный маршрут обратно маршрутизатору A. Получив объявление маршрута, маршрутизатор A отвергает его, т.к. находит в соответствующем пути себя.

Так же работает протокол BGP, за исключением того, что информация добавляется в путь к сети назначения не отдельными маршрутизаторами, а автономными системами. Любой маршрутизатор, который получил маршрут, может определить наличие петли маршрутизации, проверив присутствие в пути к заданной сети назначения своей автономной системы.

Выбор пути в протоколе BGP. Протокол BGP не использует метрики для определения петель в пути, они нужны ему для управления сетевыми правилами. Другими словами, метрики могут быть использованы сетевыми администраторами для установки сетевых правил, использующихся маршрутизаторами во время выбора пути. Протокол BGP объявляет всем своим соседям только один оптимальный маршрут. Ниже приведен список метрик, упорядоченный по возрастанию значимости:

- административный вес;
- локальное предпочтение;
- локально созданные маршруты;
- кратчайший AS-путь;
- атрибут MED (Multiple Exit Discriminator) — может использоваться только при сравнении путей, полученных от разных партнеров из одной AS; многие реализации предоставляют возможность сравнения MED для различных автономных систем. Атрибут MED будет использоваться только в конце процесса выбора лучшего маршрута;
- предпочтительные внешние пути;
- путь через ближайшего соседа, если включена синхронизация;
- путь через соседа с наименьшим идентификатором маршрутизатора [16, 17].

В большинстве сложных протоколов маршрутизации есть специальная система обнаружения соседей. С помощью этой системы маршрутизатор без труда находит своих соседей и обменивается с ними информацией о маршрутах. Протокол BGP является исключением, т.к. не занимается автоматическим обнаружением соседей и требует ручной настройки взаимоотношений между ними.

Протокол BGP не предъявляет никаких требований к топологии сети. Принцип его действия предполагает, что маршрутизация внутри автономной системы выполняется с помощью внутренних протоколов маршрутизации или, как их еще называют, интра-протоколов. Intra, что означает «внутренний», обозначает все, что относится к действиям внутри субъекта, а термин inter («внешний») означает события или действия, которые имеют место между субъектами.

Внешний протокол BGP. BGP-соседи из разных автономных систем автоматически формируют друг с другом соседские взаимоотношения на базе внешнего протокола (E-BGP, ExteriorBGP).

Маршрутизаторы-соседи обмениваются небольшими сообщениями, подтверждающими их активность на данный момент времени (keep-alive messages). Если сосед перестает получать подобные сообщения в течение некоторого предопределенного времени жизни маршрутов (hold time), он корректирует свою маршрутную таблицу, отражая в ней потерю части доступных маршрутов. Кроме того, протокол BGP4 рассылает частичные изменения, когда те или иные маршруты становятся недоступными. Таким образом, обмен полными маршрутными таблицами имеет место лишь тогда, когда два соседних маршрутизатора впервые устанавливают одноранговые отношения или когда такие отношения приходится переустанавливать повторно.

BGP4 выбирает тот маршрут, который проходит через наименьшее число автономных систем. Когда сообщение об обновлении маршрутной информации проходит через шлюз очередной автономной системы, BGP4 добавляет адрес ASN (Abstract Syntax Notation — уникальный номер AS присваивается каждой AS) этой AS к цепочке адресов других автономных систем, через которые это сообщение прошло. По умолчанию маршрут с наименьшим числом адресов ASN хранится в маршрутной таблице в качестве оптимального пути к сети назначения. Одна автономная система может содержать множество внутренних маршрутизаторов, так что фактическое число переходов, как правило, всегда больше, чем указано в строке с адресами автономных систем.

На рис. 5.39 изображен принцип работы протокола E-BGP. Маршрутизатор A объявляет префикс 10.1.1.0/24 через протокол внутреннего шлюза (IGP, InteriorGatewayProtocol) маршрутизатору B, у которого установлены соседские взаимоотношения с E-BGP-маршрутизатором C. Маршрутизатор B может преобразовать этот маршрут в маршрут протокола BGP несколькими способами.

Стандартное преобразование маршрутов. Маршрутизатор B может преобразовать маршруты IGP, использующегося между маршрутизаторами A и B, в маршруты BGP. Это приведет к маркировке

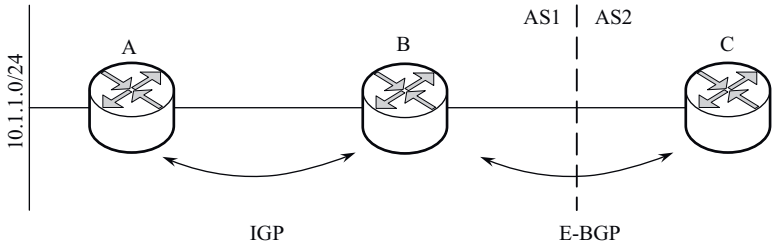


Рис. 5.39. E-BGP-соседи

преобразованных путей как «неизвестных».

Команда network. Для объявления маршрута к сети назначения 10.1.1.0/24 в маршрутизаторе B может быть использована команда network в рамках команды routerBGP. В отличие от других протоколов маршрутизации, команда network BGP не указывает, для какого интерфейса используется этот протокол, указывая вместо этого только объявляемые префиксы. Если запись в таблице маршрутизации BGP-маршрутизатора полностью совпадает (включая длину префикса) с указанным в команде network значением, то маршрутизатор объявляет этот префикс.

Команда aggregate-address. Маршрутизатор A может объединять сеть 10.1.1.0/24 в более крупный блок IP-адресов с помощью команды routerBGP.

Как только маршрутизатор B определит, что ему нужно объявить префикс маршрутизатору C, он отправит ему соответствующее обновление информации о маршруте. Список устройств до нужной подсети (AS_PATH) в этом обновлении пустой, потому что маршрут начинается в автономной системе маршрутизатора B. Адресом следующей передачи пакета для этого маршрута является IP-адрес маршрутизатора B [18, 19].

Когда маршрутизатор C получает обновление информации о маршруте, он определяет, что обновление пришло от E-BGP-соседа, добавляет номер автономной системы этого соседа в начало AS-пути и помешает префикс в таблицу маршрутизации BGP. Маршрутизатор C может вносить, а может и не вносить этот префикс в свою таблицу маршрутизации, так как это зависит от других маршрутов к данному префиксу и прочих факторов [18, 19].

Внутренний протокол (I-BGP, InternalBGP). Когда маршрутизатор установил отношения соседства с другим BGP-маршрутизатором в одной и той же автономной системе, они становятся I-BGP-соседями. Для примера рассмотрим рис. 5.40, на котором изображены I-BGP-соседи.

Как показано на рис. 5.40, маршрутизатор A объявляет сеть

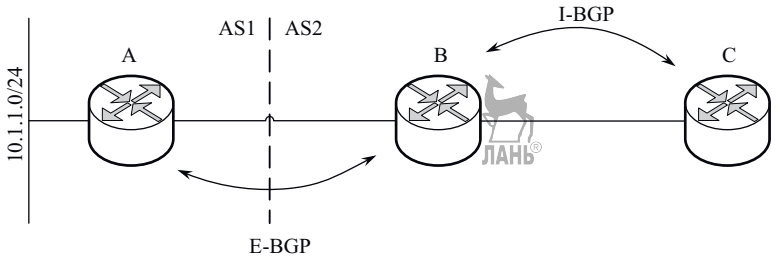


Рис. 5.40. I-BGP-соседи

10.1.1.0/24 маршрутизатору B как E-BGP-маршрут. В свою очередь, маршрутизатор B объявляет этот маршрут маршрутизатору C посредством протокола I-BGP.

Когда маршрутизатор C получает префикс 10.1.1.0/24, он не изменяет значение адреса следующей передачи пакета, а также не меняет AS-путь (потому что префикс не пересекал границу автономной системы). То, что не изменяется A, иллюстрирует одно из строгих ограничений протокола I-BGP: I-BGP-соседи не могут объявлять полученный по нему маршрут другим I-BGP-соседям. Для того, чтобы лучше понять необходимость полного объединения I-BGP-соседей, рассмотрим рис. 5.41.

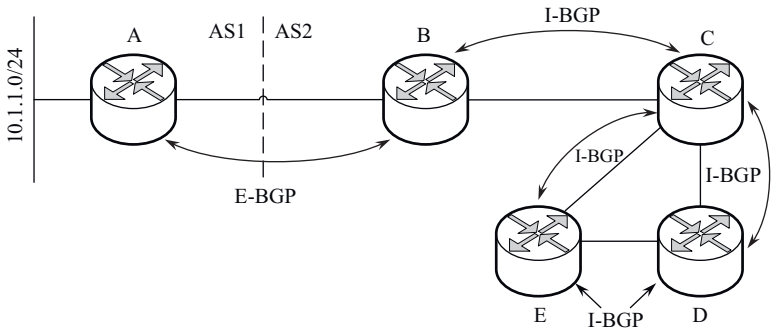


Рис. 5.41. Необходимость полного объединения I-BGP-соседей

Рассмотрим цепь событий, которые произойдут, если маршрутизатор A объявит сеть 10.1.1.0/24 маршрутизатору B. Маршрутизатор B объявляет префикс маршрутизатору C, который, в свою очередь, объявляет его маршрутизатору D. Последний объявляет префикс каждому из своих соседей, включая маршрутизатор E, который объявляет этот префикс маршрутизатору C. К этому времени маршрутизатор C получил два I-BGP-объявления префикса 10.1.1.0/24 — одно от маршрутизатора B и одно от маршрутизатора E.

Так как значение адреса следующей передачи пакета и AS-путь

не изменились при объявлении префикса от одного соседа другому, маршрутизатор C не может определить, что путь, полученный от маршрутизатора E, является петлей.

Чтобы предотвратить проблемы такого рода, I-BGP-соседи не объявляют маршрут, полученный одним I-BGP-соседом от другого. Однако на самом деле это означает, что I-BGP-соседи должны быть полностью объединены.

Формат заголовка сообщения протокола BGP. Формат заголовка сообщения в BGP представляет собой поле маркера длиной 16 байт, за которым следует поле длины (2 байта) и поле типа (1 байт). На рис. 5.42 представлен формат заголовка сообщения протокола BGP.

Рис. 5.42. Формат заголовка сообщения BGP

В зависимости от типа сообщения в сообщении протокола BGP за заголовком может следовать или не следовать блок данных.

Поле маркера длиной 16 байт используется для аутентификации входящих сообщений BGP, либо для детектирования потери синхронизации между двумя взаимодействующими по BGP маршрутизаторами [3, 19].

5.6. Протокол резервирования ресурсов — RSVP

Задача протокола RSVP (Resource Reservation Protocol) заключается в том, чтобы ЭВМ могла запросить для приложения определенный уровень качества сетевых услуг QoS (например, определенный уровень полосы пропускания). RSVP используется также маршрутизаторами для доставки QoS-запросов всем узлам вдоль пути информационного потока, а также для установки и поддержания необходимого уровня услуг. RSVP-запросы обеспечивают резервирование определенных сетевых ресурсов, которые нужны, чтобы обеспечить конкретный уровень QoS вдоль всего маршрута транспортировки данных [3, 13].

RSVP имеет следующие атрибуты:

– выполняет резервирование для уникастных и мультикастных приложений, динамически адаптируясь к изменениям член-

ства в группе вдоль маршрута;
- является симплексным протоколом, т.е. он выполняет резервирование для однонаправленного потока данных;
- ориентирован на получателя, т.е. получатель данных инициирует и поддерживает резервирование ресурсов для потока;
- поддерживает динамическое членство в группе и автоматически адаптируется к изменениям маршрутов;
- не является маршрутным протоколом, но зависит от существующих и будущих маршрутных протоколов;
- транспортирует и поддерживает параметры управления трафиком и политикой, которые остаются непрозрачными для RSVP;
- обеспечивает несколько моделей резервирования или стилей, чтобы удовлетворить требованиям различных приложений;
- обеспечивает прозрачность операций для маршрутизаторов, которые его не поддерживают;
- может работать с IPv4 и IPv6 [14, 20].

Чтобы обеспечить должное качество обслуживания трафика речевых и видео-приложений, необходим механизм, позволяющий приложениям информировать сеть о своих требованиях. На основе этой информации сеть может резервировать ресурсы для того, чтобы гарантировать выполнение требований к качеству, или отказать приложению, вынуждая его либо пересмотреть требования, либо отложить сеанс связи. В роли такого механизма выступает протокол резервирования ресурсов RSVP (Resource Reservation Protocol).

В основе протокола RSVP лежат три компонента:
- *сеанс связи*, который идентифицируется адресом получателя данных;
- *спецификация потока*, которая определяет требуемое качество обслуживания и используется узлом сети, чтобы установить соответствующий режим работы диспетчера очереди;
- *спецификация фильтра*, определяющая тип трафика, для обслуживания которого запрашивается ресурс.

Процесс резервирования ресурсов можно разбить на пять отдельных шагов:

1. Отправитель данных передает на индивидуальный или групповой адрес получателя сообщение Path, в котором указывает желательные характеристики качества обслуживания трафика — верхнюю и нижнюю границу полосы пропускания, величину задержки и вариации задержки. Данные сообщения Path передаются по тому же пути, по которому они отправляют обычный трафик с данными. В этих сообщениях описываются данные, которые уже отправляются

или только будут отправляться.

2. Каждый маршрутизатор, поддерживающий протокол RSVP, получив сообщение Path, фиксирует определенный элемент «структуры пути» — адрес предыдущего маршрутизатора. Таким образом, в сети образуется фиксированный маршрут. Поскольку сообщения Path содержат те же адреса отправителя и получателя, что и данные, пакеты будут маршрутизироваться корректно даже через сетевые области, не поддерживающие протокол RSVP.

3. Станции-получатели выбирают подмножество сеансов, для которых они получили PATH-информацию и с помощью RSVP RESV-сообщения запрашивают RSVP-резервирование ресурсов у предыдущего маршрутизатора. RSVP RESV-сообщения идут от получателя к отправителю в противоположном направлении по маршруту, пройденному RSVP PATH-сообщениями.

4. RSVP-маршрутизаторы определяют, могут ли они удовлетворить эти RESV-запросы. Если нет, они отказывают в резервировании. Если да, то они объединяют полученные запросы на резервирование и отсылают запрос предыдущему маршрутизатору.

5. Отправители, получив запросы на резервирование ресурсов от соответствующих маршрутизаторов, считают резервирование ресурсов состоявшимся, то есть реальное резервирование ресурсов осуществляется RESV-сообщениями [20].

Механизм RSVP-резервирования схематически показан на рис. 5.43.

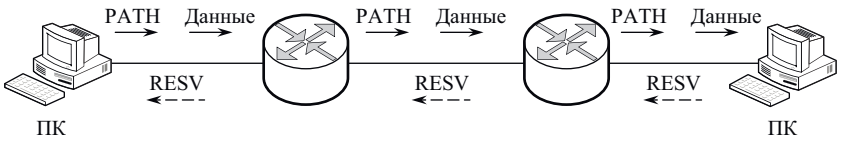


Рис. 5.43. Резервирование ресурсов

Отменить резервирование можно двумя путями: прямо или косвенно. В первом случае отмена производится по инициативе отправителя или получателя с помощью специальных сообщений RSVR. Во втором случае резервирование отменяется по таймеру, ограничивающему срок существования резервирования.

Несмотря на то, что протокол RSVP является важным инструментом в арсенале средств, обеспечивающих гарантированное качество обслуживания, этот протокол не может решить все проблемы, связанные с QoS. Основные недостатки протокола RSVP — большой объем служебной информации и большие затраты времени на организацию резервирования.

Протокол RSVP не используется в крупномасштабных средах.

В лучшем случае магистральный маршрутизатор имеет возможность резервировать ресурсы для нескольких тысяч потоков и управлять очередями для каждого из них [8, 20].

5.7. Протокол передачи RTP (Real-Time Transport Protocol)

Стремительный рост Internet предъявляет новые требования к скорости и объемам передачи данных. И для того, чтобы удовлетворить все эти запросы, одного увеличения емкости сети недостаточно, необходимы разумные и эффективные методы управления трафиком и контролем загруженности линий передачи.

Эту задачу и призван решить новый транспортный протокол реального времени (RTP), который работает поверх UDP (см. рис. 5.44) и гарантирует доставку данных одному или более адресатам с задержкой в заданных пределах, т.е. данные могут быть воспроизведены в реальном масштабе времени и предназначены для передачи видео- и аудиоинформации.

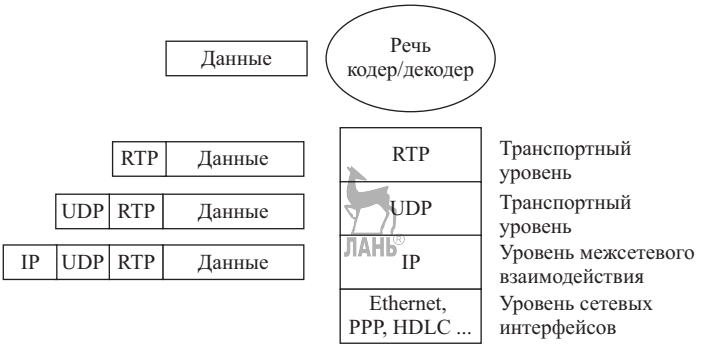


Рис. 5.44. Место RTP в стеке протоколов TCP/IP

В протоколе предусмотрены следующие функции:

1. Идентификация отправителя — каждый RTP-пакет содержит идентификатор отправителя, указывающий, кто из участников генерирует данные.

2. Идентификация типа полезной нагрузки — специальное поле идентифицирует формат трафика RTP и определяет его интерпретацию приложением.

3. Определение порядка и момента декодирования каждого пакета. На стороне-отправителе каждому исходящему пакету присваивается временная метка и порядковый номер. На принимающей стороне эти данные указывают на то, в какой последовательности и с какими задержками их необходимо воспроизводить, а также позволяют интерполировать потерянные пакеты.

4. Обнаружение потерянных пакетов — порядковые номера делают возможным и это.

5. Синхронизация — использование временных меток делает возможным синхронное воспроизведение мультимедийных данных.

Протокол реализует распознавание типа трафика, нумерацию последовательности пакетов, работу с метками времени и контроль передачи. Данный протокол регламентирует передачу мультимедийных данных в пакетах через информационно-вычислительные сети на транспортном уровне и дополняется протоколом управления передачей данных в реальном масштабе времени RTCP (Real-Time Control Protocol). Протокол RTCP, в свою очередь, обеспечивает контроль доставки мультимедийных данных, контроль качества обслуживания, передачу информации об участниках текущего сеанса связи, управление и идентификацию, и иногда считается частью протокола RTP.

Таким образом, действие протокола RTP сводится к присваиванию каждому исходящему пакету временных меток. На приемной стороне временные метки пакетов указывают на то, в какой последовательности и с какими задержками их необходимо воспроизводить. Поддержка RTP и RTCP позволяет принимающему узлу располагать принимаемые пакеты в надлежащем порядке, снижать влияние неравномерности времени задержки пакетов в сети на качество сигнала и восстанавливать синхронизацию между аудио и видео, чтобы поступающая информация могла правильно прослушиваться и просматриваться пользователями [8, 21].

RTP сам по себе не имеет никакого механизма, гарантирующего своевременную передачу данных и качество обслуживания, но для обеспечения этого использует службы нижележащего уровня. Он не предотвращает нарушения порядка следования пакетов, но при этом и не предполагает, что основная сеть абсолютно надежна и передает пакеты в нужной последовательности. Порядковые номера, включенные в RTP, позволяют получателю восстанавливать последовательность пакетов отправителя.

Протокол RTP поддерживает как двустороннюю связь, так и передачу данных группе адресатов, если групповая передача поддерживается нижележащей сетью. RTP предоставляет информацию, требуемую отдельным приложениям, и в большинстве случаев интегрируется в работу приложения.

Хотя протокол RTP считается протоколом транспортного уровня, он функционирует обычно поверх другого протокола транспортного уровня UDP (User Datagram Protocol). Оба протокола вносят свои доли в функциональность транспортного уровня. Следует отметить, что RTP и RTCP являются независимыми от нижележащих уровней —

транспортного и сетевого, поэтому протоколы RTP/RTCP могут использоваться с другими подходящими транспортными протоколами.

Протокольные блоки данных RTP/RTCP называются пакетами. Пакеты, формируемые в соответствии с протоколом RTP и служащие для передачи мультимедийных данных, называются информационными пакетами или пакетами данных (data packets), а пакеты, генерируемые в соответствии с протоколом RTCP и служащие для передачи служебной информации, требуемой для надежной работы телеконференции, называют пакетами управления или служебными пакетами (control packets).

Таким образом, полная спецификация RTP для конкретного приложения должна включать дополнительные документы, к которым относятся описание профиля, а также описание формата трафика, определяющее, как трафик конкретного типа, такой как аудио или видео, будет обрабатываться в RTP [8].

Формат заголовка протокола RTP. RTP — потоко-ориентированный протокол. Заголовок RTP-пакета содержит информацию о порядке следования пакетов, чтобы поток данных был правильно собран на принимающем конце, и временную метку для правильного чередования кадров при воспроизведении и для синхронизации нескольких потоков данных, например, видео и аудио.

Каждый пакет RTP имеет основной заголовок, а также, возможно, дополнительные поля, специфичные для приложения.

Использование TCP в качестве транспортного протокола для этих приложений невозможно по нескольким причинам:

- этот протокол позволяет установить соединение только между двумя конечными точками, следовательно, он не подходит для многоадресной передачи;
- brTCP предусматривает повторную передачу потерянных сегментов, прибывающих, когда приложение реального времени уже их не ждет;
- TCP не имеет удобного механизма привязки информации о синхронизации к сегментам — дополнительное требование приложений реального времени.

Несмотря на то, что каждое приложение реального времени может иметь свои собственные механизмы для поддержки передачи в реальном времени, они имеют много общих черт, а это делает определение единого протокола весьма желательным.

Эту задачу и призван решить новый транспортный протокол реального времени — RTP (Real-time Transport Protocol), который гарантирует доставку данных одному или более адресатам с задержкой в заданных пределах, т.е. данные могут быть воспроизведены в ре-

ального времени.

В общем виде распределение протоколов по уровням модели OSI выглядит следующим образом:

– транспортный уровень: RTP поверх UDP;
– сетевой IP;
– канальный: Ethernet;
– физический: Ethernet.

При передаче информации с использованием протокола RTP используется инкапсуляция следующего вида (рис. 5.45).

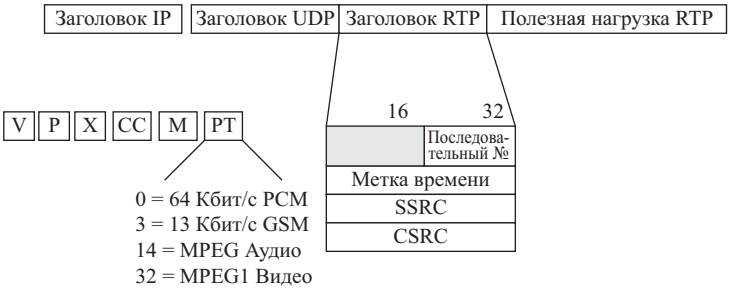


Рис. 5.45. Инкапсуляция протокола RTP

Минимальный размер сегмента RTP составляет 12 байт. На рис. 5.46 представлен фиксированный RTP-заголовок, который содержит ряд полей, идентифицирующих такие элементы, как формат пакета, порядковый номер, источники, границы и тип полезной нагрузки. За фиксированным заголовком могут следовать другие поля, содержащие дополнительную информацию о данных.

V	P	X	CC	M	PT	Sequence Number
Synchronization Source (SSRC) Identifier						
Times tamp						
Contributing Source (CSRC) Identifiers						

Рис. 5.46. Фиксированный RTP-заголовок

V (2 бита). Поле версии. Текущая версия — вторая.

P (1 бит). Поле заполнения — сигнализирует о наличии заполняющих октетов в конце полезной нагрузки. Заполнение применяется, когда приложение требует, чтобы размер полезной нагрузки был кратен, например, 32 битам. В этом случае последний октет указывает число заполняющих октетов.

X (1 бит). Поле расширения заголовка. Когда это поле задано, то за основным заголовком следует еще один дополнительный, используемый в экспериментальных расширениях RTP.

CC (4 бита). Поле числа отправителей — содержит число идентификаторов отправителей, чьи данные находятся в пакете, причем сами идентификаторы следуют за основным заголовком.

M (1 бит). Поле маркера — зависит от типа полезной нагрузки, он используется обычно для указания границ потока данных. В случае видео он задает конец кадра. В случае голоса он задает начало речи после периода молчания.

PT (7 бит). Поле типа полезной нагрузки, оно идентифицирует тип полезной нагрузки и формат данных, включая сжатие и шифрование. В стационарном состоянии отправитель использует только один тип полезной нагрузки в течение сеанса, но он может его изменить в ответ на изменение условий, если об этом сигнализирует протокол управления передачей в реальном времени (Real-Time Transport Control Protocol).

Sequence Number (16 бит). Поле порядкового номера. Каждый источник начинает нумеровать пакеты с произвольного номера, увеличиваемого затем на единицу с каждым посланным пакетом данных RTP. Это позволяет обнаружить потерю пакетов и определить порядок пакетов с одинаковой отметкой о времени. Несколько последовательных пакетов могут иметь одну и ту же отметку о времени, если логически они порождены в один и тот же момент, как, например, пакеты, принадлежащие к одному и тому же видеокадру.

Synchronization Source (SSRC) Identifier (32 бита). Поле идентификатора источника синхронизации — генерируемое случайным образом число, уникальным образом идентифицирующее источник в течение сеанса и независимое от сетевого адреса. Это число играет важную роль при обработке поступившей порции данных от одного источника.

Time stamp (32 бита). Поле отметки о времени — содержит момент времени, в который первый октет данных полезной нагрузки был создан. Единицы, в которых время указывается в этом поле, зависят от типа полезной нагрузки. Значение определяется по локальным часам отправителя.

Contributing source (CSRC) Identifier (32 бита). Список полей идентификаторов источника, «подмешанных» в основной поток, например, с помощью микшера. Микшер вставляет целый список SSRC идентификаторов источников, которые участвовали в построении данного RTP-пакета. Этот список и называется CSRC. Количество элементов в списке от 0 до 15. Если число участников более 15, выбираются первые 15. Примером может служить аудио-конференция, в RTP-пакеты которой собраны речи всех участников, каждый со своим SSRC — они-то и образуют список CSRC. При этом вся конференция имеет общий SSRC [8, 22].

5.8. Протокол DHCP (Dynamic Host Configuration Protocol)

Для автоматического назначения IP-адресов используется протокол динамической настройки хоста — DHCP.

DHCP был разработан для того, чтобы освободить администратора от ручной работы. DHCP осуществляет не только назначение адресов по динамическому признаку, но и поддерживает способы ручного и автоматического статического назначения IP-адресов.

При ручном способе администратор представляет DHCP-серверу информацию о соответствии IP-адресов с физическим адресом или идентификатором клиента. Эти адреса сообщаются клиентам в ответ на их запросы DHCP-серверу.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес (и ряд других параметров конфигурации клиента) из пула памяти намеченных IP-адресов без вмешательства администратора. Граница адресов определяется администратором при конфигурировании DHCP-сервера.

Между идентификаторами клиента и его IP-адресом существует постоянное соответствие. Оно устанавливается в момент первичного назначения сервером DHCP IP-адреса клиенту.

При последовательных запросах сервер возвращает тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, что дает возможность повторно использовать IP-адрес другой компании.

Динамическое распределение адресов позволяет строить IP-сеть, количество узлов которой намного превышает количества имеющихся в распоряжении администраторов IP-адресов.

DHCP протокол обеспечивает надежный и простой способ конфигурирования в сети TCP/IP, гарантируя отсутствие конфликтных адресов за счет централизованного управления и распределения адресов.

Администратор управляет процессом назначения адресов с помощью параметра «продолжительность аренды», которая определяет, как долго компьютер может использовать назначенный IP-адрес перед тем, как снова запросить его от сервера DHCP в аренду.

Рассмотрим пример работы протокола DHCP. DHCP может использовать свои возможности в основном, если компьютер удаляется из подсети и IP-адрес автоматически освобождается.

Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес, при этом ни пользователь, ни администратор не вмешиваются в данный процесс назначения IP-адреса. Особенно это важно для мобильных пользователей.

DHCP использует модель клиент-сервер. Во время старта системы компьютер-клиент DHCP, находящийся в состоянии инициализации, посылает сообщение «Исследователь», которое широковещательно передается по локальной сети всем DHCP-серверам частной интерсети. Пакет «Исследователь» содержит IP-адрес 127.0.0.1.

Каждый DHCP-сервер, получивший данное сообщение, отвечает на него сообщением OFFER — предложение, которое содержит IP-адреса и информацию о конфигурации данного узла.

Компьютер DHCP переходит в состояние выбора и собирает конфигурационные предложения от DHCP-серверов, затем он выбирает одно из этих предложений, переходит в состояние «запрос» и отправляет сообщение «Request» (запрос) тому DHCP-серверу, чье предложение было выбрано. Выбранный DHCP-сервер посылает сообщение — подтверждение, которое содержит тот же IP-адрес, который уже был послан ранее на стадии исследования, а также параметр аренды этого адреса. Кроме того, DHCP-сервер посылает параметры сетевой конфигурации.

После получения клиентом этого подтверждения, он переходит в состояние связь, после чего он может принимать участие в работе сети TCP/IP.

Компьютеры и клиенты, которые имеют локальные диски, сохраняют полученный адрес для использования при последующих стартах системы. При приближении момента истечения срока аренды адреса компьютер пытается обновить параметры аренды у DHCP-сервера. Если этот IP-адрес ему не может быть выделен снова, то ему возвращается другой IP-адрес.

В DHCP-сервере описывается несколько типов сообщений:

- для обнаружения и выбора DHCP адресов;
- для запросов информации о конфигурации;
- для продления аренды IP-адреса;
- для досрочного закрытия IP-адреса.

Проблемы, вносимые DHCP протоколом:

- согласование информационно-адресной базы в службах DHCP и DNS, так как если IP-адрес будет динамически изменяться сервером DHCP, то необходимо тут же изменить базу данных DNS;
- нестабильность IP-адресов усложняет процесс управления сетью с помощью протокола SNMP;
- при централизованном назначении адресов в случае отказа DHCP-сервера клиенты оказываются не в состоянии получить IP-адрес, следовательно, приходится дополнительно устанавливать DHCP-сервера [8, 23].

Взаимодействие клиента и сервера при выделении сетевого адреса. Протокол DHCP использует четырехэтапный процесс для конфигурации своего клиента. Если у компьютера несколько сетевых адаптеров, то каждый из них конфигурируется отдельно и ему назначается уникальный IP-адрес. Передача данных между DHCP-клиентом и DHCP-сервером происходит по UDP через порты 67 и 68.

Большинство сообщений протокола DHCP передаются с использованием широковещания. Для связи DHCP-клиентов с DHCP-сервером в удаленной сети IP-маршрутизаторы должны поддерживать ретрансляцию широковещательных сообщений DHCP. В табл. 5.4 и на рис. 5.47 отображены этапы конфигурирования протокола DHCP.

Таблица 5.4. Этапы конфигурирования протокола DHCP

Этап	Описание
Запрос аренды IP-адреса	Клиент инициализирует ограниченную версию протокола TCP/IP и посылает широковещательный запрос для поиска DHCP-сервера и информации об IP-адресации
Предложение аренды	Все серверы протокола DHCP, имеющие свободную информацию об IP-адресах, отправляют предложение клиенту
Выбор аренды	Клиент выбирает информацию об IP-адресации из первого полученного предложения и посылает широковещательное сообщение с запросом информации об аренде IP-адреса
Подтверждение аренды DHCP	Сервер, сделавший это предложение, отвечает на запрос, а все остальные серверы отзывают свои предложения. Клиенту назначается IP-адрес и сопутствующие параметры. Клиент завершает настройку и связывает TCP/IP с остальными компонентами системы. Поскольку автоматическая конфигурация выполнена, клиент может использовать все сервисы и утилиты протокола TCP/IP для связи с другими узлами TCP/IP

5.9. Протокол LDAP

LDAP (Lightweight Directory Access Protocol) — упрощенный протокол доступа к каталогам. Используется для динамического присвоения IP-адресов пользователям и обеспечения доступа к сетевым ресурсам. Является стандартом доступа к службам сетевых каталогов.

Для объединения двух каналов необходимо установить 2 сервера (рис. 5.48).

Протокол LDAP упрощает работу в сетевой среде, так пользователь получает возможность входить в систему с любого узла сети и работать с привычными для себя надстройками, т.к. информация

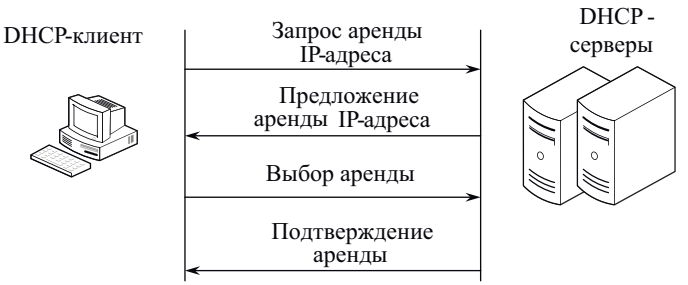


Рис. 5.47. Этапы конфигурирования протокола DHCP

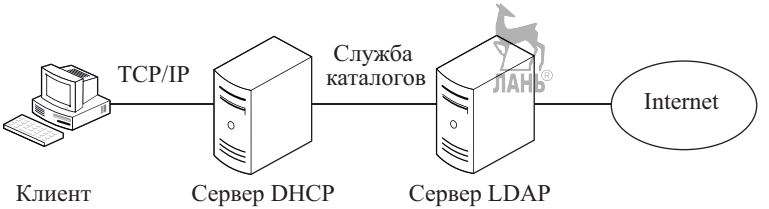


Рис. 5.48. Схема объединения двух серверов

о них будет сохраняться на сервере LDAP в каталогах. При этом не только DHCP, но и DNS будет использовать каталоги с сервера LDAP в качестве своих хранилищ информации, тогда эти службы будут иметь дополнительное достоинство: это модельная структура и независимость от места размещения.

Протокол LDAP предназначен специально для использования с управляющими и браузерными приложениями, которые обеспечивают интерактивный доступ к каталогам с возможностью чтения и записи.

Работает протокол LDAP поверх протокола TCP/IP и обеспечивает доступ к службам каталогов, используемых сети Internet или других служб.

Совмещение протоколов DNS и DHCP на базе протокола LDAP обеспечивают:

- доступ к информации, позволяющий организовать поиск и сохранение данных в информационных базах данных с использованием хранилищ серверов DHCP и DNS;
- гибкость построения сети, т.к. сетевой протокол LDAP может работать на различных платформах, следовательно, можно разместить хранилище информации на других машинах;
- организацию работы одного или нескольких пользователей с одним и тем же документом или базой данных.

Главная цель объединения серверов — дать возможность пользователям встраивать в их систему управления сетевыми адресами средства, которые повышают надежность, безопасность, а главное —

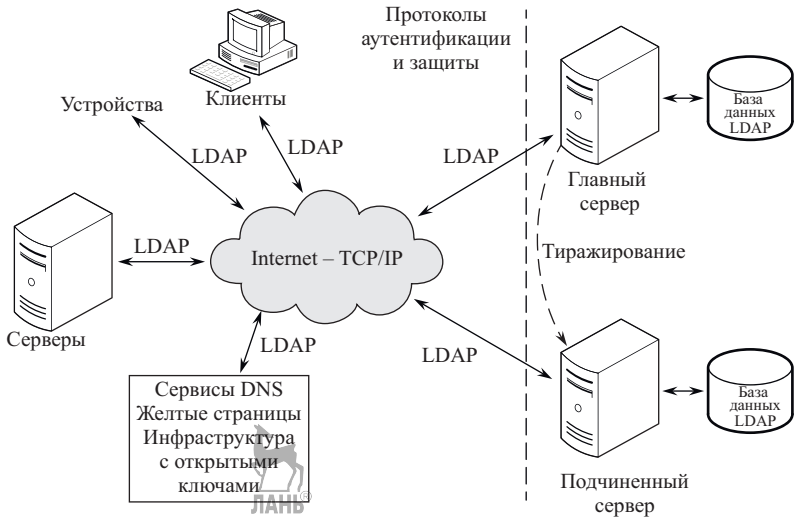


Рис. 5.49. Инфраструктура LDAP. Устройства и серверы с помощью протокола LDAP обращаются к данным, хранимым в базе данных LDAP-серверов

синхронизацию имен и адресов.

Принцип взаимодействия адресов: клиент посылает запрос на доступ в Internet с указанием нужного адреса, а также ресурса. Сервер DHCP автоматически присваивает клиенту IP-адрес и связывает пользователя с ресурсами в каталоге LDAP. Сервер LDAP находит указанные ресурсы и автоматически соединяет пользователя с соответствующими узлами сети. Как и DNS, LDAP — это служба каталогов в архитектуре клиент-сервер. Каталоги могут содержать самую разную информацию, например, базу данных пересчета телефонных номеров в IP-адрес для пользователя IP-телефонии. LDAP хранит данные на нескольких серверах; если при обращении клиента LDAP для выбора адреса шлюза IP-телефонии сервер не может ответить на запрос, то он возвращает клиенту указатель на другой сервер LDAP, где информация может быть найдена (см. рис. 5.49) [3, 24].

5.10. Протоколы ARP, RARP

Назначение протокола ARP (Address Resolution Protocol). Любое устройство, подключенное к локальной сети (Ethernet, FDDI и т.д.), имеет уникальный физический MAC-адрес, заданный аппаратным образом. 6-байтовый Ethernet-адрес выбирает изготовитель сетевого интерфейсного оборудования из выделенного для него по лицензии адресного пространства. Если у машины меняется сетевой адаптер, то меняется и ее Ethernet-адрес [3].

4-байтовый IP-адрес задает менеджер сети с учетом положения машины в сети Интернет. Если машина перемещается в другую часть сети Интернет, то ее IP-адрес должен быть изменен. Преобразование IP-адресов выполняется с помощью ARP-таблицы. Каждая машина сети имеет отдельную ARP-таблицу для каждого своего сетевого адаптера. Поэтому, существует проблема отображения физического адреса (6 байт для Ethernet) в пространство сетевых IP-адресов (4 байта) и наоборот [3, 25].

Протокол ARP (Address Resolution Protocol), рассмотренный в рекомендации RFC-826, преобразует IP-адреса в MAC-адреса [3, 25].

Объекты протокола ARP классифицируются либо как клиенты разрешения адреса (Address Resolution Clients), либо как услуги разрешения адреса (Address Resolution Services). Клиенты разрешения адреса обычно реализуются в узлах клиентов, в то время как услуги разрешения адреса обычно обеспечиваются узлами обслуживания [15, 25].

Протокол определяет, передавать ли данные через сетевой уровень к верхним уровням эталонной модели OSI. Для передачи данных к верхним уровням необходимо, чтобы пакет данных содержал MAC- и IP-адрес пункта назначения. Если в пакете данных отсутствует один из этих адресов, данные не будут переданы на верхние уровни. Таким образом, MAC- и IP-адрес служат для своего рода проверки и дополнения друг друга.

Протокол ARP устанавливает соответствие между сетевыми и локальными адресами либо на основании заранее составленных таблиц, либо путем рассылки широковещательных запросов [4, 25]. Таблица соответствия локальных адресов сетевым адресам строится отдельно для каждого сетевого интерфейса. Протокол разрешения адресов занимает промежуточное положение между сетевым и канальным уровнями.

Формат протокола ARP. В отличие от большинства протоколов, данные в пакетах ARP не имеют фиксированного формата заголовка. Сообщения были разработаны так, чтобы их можно было использовать для различных сетевых технологий. Функционально ARP делится на две части: одна определяет физический адрес при посылке пакета, другая отвечает на запросы других машин. ARP-пакеты вкладываются непосредственно в Ethernet-кадры (см. рис. 5.50).

Поле *«Тип оборудования»* — это тип интерфейса, для которого отправитель ищет адрес.

Поле *«Тип протокола»* содержит код типа протокола (для IP это 0800H).

Поле *«HA-Len»* — длина аппаратного адреса.

0	8	16	24	31
Тип оборудования		Тип протокола		
HA-len	PA-len	Код операции		
Аппаратный адрес отправителя				
Адрес отправителя		IP-адрес отправителя		
IP-адрес отправителя		Аппаратный адрес адресата		
Аппаратный адрес адресата				
IP-адрес адресата				

Рис. 5.50. Формат пакета ARP

Поле *«PA-Len»* — длина протокольного адреса (длина в байтах, например, для IP-адреса PA-Len=4).

Поле *«Код операции»* определяет, является ли данный пакет ARP-запросом (код = 1), ARP-откликом (2), RARP-запросом (3), или RARP-откликом (4) [9, 25].

Работа протокола ARP. Протокол ARP работает различным образом в зависимости от того, какой протокол канального уровня используется в данной сети — протокол локальной сети (Ethernet, FDDI) с возможностью широковещательного доступа одновременно ко всем узлам сети или же протокол глобальной сети (X.25, Frame Relay), как правило, не поддерживающий широковещательный доступ.

Необходимость в обращении к протоколу ARP возникает каждый раз, когда модуль IP передает пакет на уровень сетевых интерфейсов, например, драйверу Ethernet. IP-адрес узла назначения известен модулю IP. Требуется на его основе найти MAC-адрес узла назначения.

Единственный способ установления соответствия — ведение таблиц. В результате конфигурирования сети каждый интерфейс знает свой IP-адрес и MAC-адрес, что можно рассматривать как таблицу, состоящую из одной строки.

Когда отправитель определил IP-адрес получателя (рис. 5.51), он смотрит в свою ARP-таблицу (табл. 5.5), чтобы узнать его MAC-адрес. Если источник обнаруживает, что MAC- и IP-адрес получателя присутствуют в его таблице, он устанавливает соответствие между ними, а затем использует их в ходе инкапсуляции данных. После этого пакет данных по сетевой среде отправляется адресату.

Рассмотрим работу протокола ARP в локальных сетях с широковещанием (рис. 5.52).

Обмен имеющейся информацией между узлами сети и реализуется ARP протоколом.

На рис. 5.52 рассмотрен фрагмент IP-сети, включающей две сети Ethernet, каждая из них подключена к интерфейсам 1 и 2 маршрутизатора. Каждый интерфейс имеет MAC- и IP-адрес. Пусть в опре-

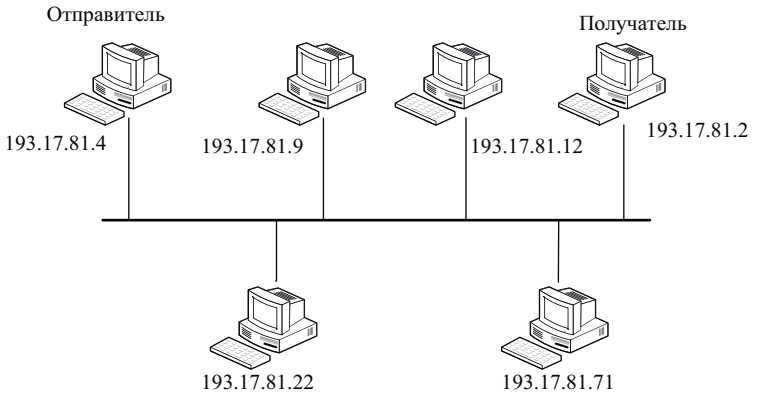


Рис. 5.51. Источник сверяется со своей ARP-таблицей после того, как определит IP-адрес пункта назначения

Таблица 5.5. Пример ARP-таблицы

IP-адрес	MAC-адрес	Тип записи
193.17.81.4	008048EB7E61	Динамический
193.17.81.9	08007A21A722	Динамический
193.17.81.2	008048EB5567	Статический
193.17.81.22	08005A21A283	Динамический
193.17.81.71	08005A21A237	Статический
193.17.81.12	00805A21B284	Динамический

деленный момент времени IP-модуль узла C направляет пакет узлу D. Протокол IP узла C определяет IP-адрес интерфейса следующего маршрутизатора — это IP_1. Далее, прежде чем упаковать пакет в кадр Ethernet и направить его маршрутизатору, необходимо определить соответствующий MAC-адрес. Для решения этой задачи протокол IP обращается к протоколу ARP. Протокол ARP поддерживает на каждом интерфейсе сетевого адаптера или маршрутизатора отдельную ARP-таблицу, в которой в ходе функционирования сети накапливается информация о соответствии между IP-адресами и MAC-адресами других интерфейсов данной сети. Первоначально при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

1. На первом шаге происходит передача от протокола IP к протоколу ARP, используя, например, сообщение «Какой MAC-адрес имеет интерфейс с адресом IP_1?».

2. ARP просматривает ARP-таблицу соответствующего интерфейса. Допустим, что необходимой записи с IP_1-адресом нет.

3. В этом случае исходящий IP-пакет, для которого оказалось невозможным определить локальный адрес из ARP-таблицы, запоминается в буфере, а протокол ARP формирует ARP-запрос, вкладывает

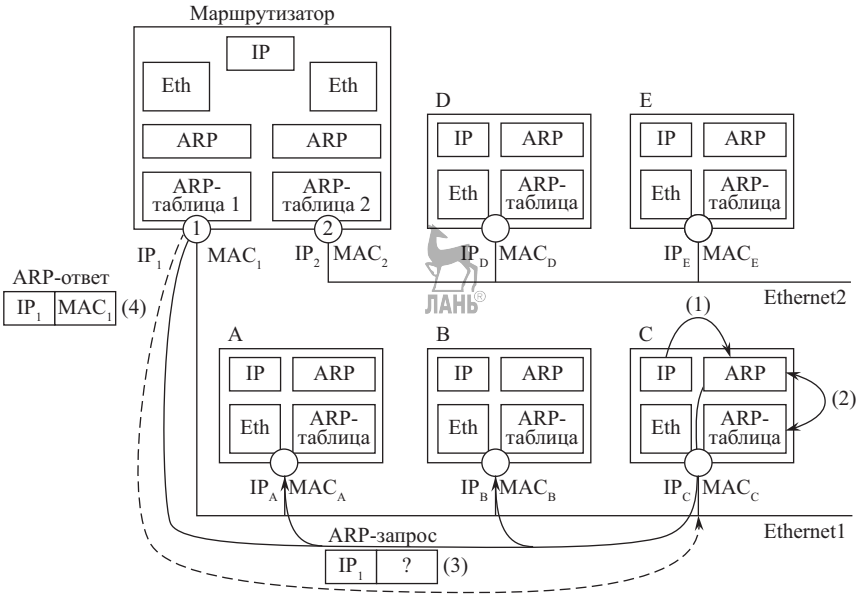


Рис. 5.52. Схема работы протокола ARP

его в кадр Ethernet и широковещательно рассылается.

4. Все интерфейсы сети Ethernet 1 получат ARP-запрос и направят его своему протоколу ARP. ARP сравнивает указанный в запросе адрес IP_1 с IP-адресом интерфейса, на который поступил данный запрос. Протокол ARP, который определил совпадение (в этом случае ARP маршрутизатора 1), формирует ARP-ответ.

В ARP-ответе маршрутизатор указывает локальный MAC-адрес своего интерфейса и отправляет его запрашивающему узлу (на рис. 5.52 это узел C), используя его локальный адрес. Широковещательный ответ в этом случае не требуется, так как формат ARP-запроса предусматривает поля локального и сетевого адресов отправителя. В данном случае необходимо подчеркнуть, что зона действия распространения ARP-запросов ограничивается сетью Ethernet 1, так как на пути широковещательных кадров стоит барьером маршрутизатор.

Протокол RARP (Reverse Address Resolution Protocol). Для того, чтобы сетевое устройство могло отправить данные на транспортный уровень эталонной модели OSI, необходимы и MAC-, и IP-адрес. Таким образом, MAC- и IP-адрес служат для проверки и дополнения друг друга. Чтобы получатель, принимающий данные, знал, кто их отправил, пакет данных должен содержать MAC- и IP-адреса источника.

А что произойдет, если источник знает свой MAC-адрес, но не знает своего IP-адреса? Протокол, который используют устройства, не знающие своего IP-адреса, называется протоколом обратного преобразования адреса (RARP).

Представим, что источник хочет послать данные другому устройству. Однако источник знает свой MAC-адрес, но не может обнаружить собственный IP-адрес в своей ARP-таблице. Чтобы получатель мог оставить у себя данные, передать их на верхние уровни эталонной модели OSI и распознать устройство, которое отправило данные, источник должен включить в пакет данных свои MAC- и IP-адреса. Поэтому источник инициирует процесс, называемый RARP-запросом, позволяющий ему определить собственный IP-адрес. Для этого устройство создает пакет RARP-запроса и посылает его в сеть. Для того чтобы пакет RARP-запроса был замечен всеми устройствами в сети, источник использует IP-адрес широковещания.

RARP-запросы имеют такую же структуру, как и ARP-запросы. RARP-запрос состоит из MAC- и IP-заголовка, а также сообщения RARP-запроса. Единственное отличие в формате RARP-пакета заключается в том, что заполнены MAC-адреса источника и получателя, а поле IP-адреса источника — пустое. Поскольку сообщение передается в режиме широковещания, то есть всем устройствам в сети, адрес назначения записывается в виде всех двоичных единиц.

Так как RARP-запрос посылается в режиме широковещания, его видят все устройства в сети. Однако только специальный RARP-сервер может отозваться на RARP-запрос. RARP-сервер служит для отправки RARP-ответа, в котором содержится IP-адрес устройства, создавшего RARP-запрос.

RARP-ответы имеют такую же структуру, как и ARP-ответы. RARP-ответ состоит из сообщения RARP-ответа, MAC- и IP-заголовка. Когда устройство, создавшее RARP-запрос, получает ответ, оно обнаруживает свой IP-адрес [3, 26].

5.11. Протокол TCP (Transmission Control Protocol)

Особенности работы протокола TCP. Несмотря на кажущуюся простоту, TCP протокол достаточно сложен и должен решать следующие основные проблемы:

- восстанавливать порядок сегментов;
- убирать дубликаты сегментов, в каком бы виде (фрагментация) они не поступали;
- определять разумную задержку для «timeout» для подтверждений в получении сегмента;
- устанавливать и разрывать соединения надежно;

– управлять потоком;
– управлять перегрузками.

Заголовок TCP. Заголовок TCP, который помещается в модуль PDU (Protocol Data Unit — протокольная единица обмена), представлен на рис. 5.53. Каждый сегмент начинается с 20-байтового заголовка фиксированного формата. За фиксированным заголовком могут следовать дополнительные поля. После дополнительных полей может располагаться до $65\,536 - 20 - 20 = 65\,496$ байт данных, где первые 20 байт означают IP-заголовок, а вторые — TCP-заголовок. Сегменты могут и не содержать данных. Такие сегменты часто применяются для передачи подтверждений и управляющих сообщений [8, 27].

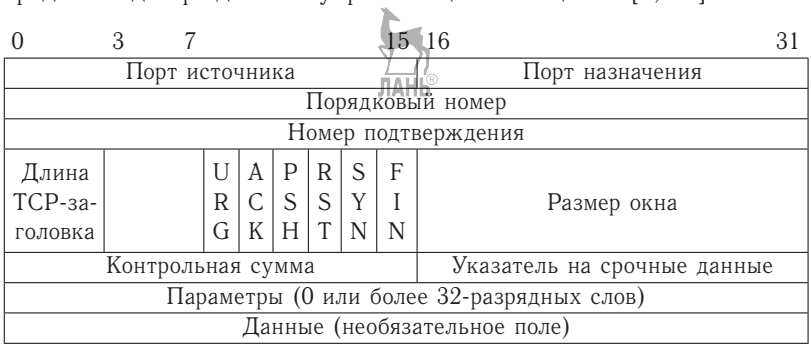


Рис. 5.53. TCP-заголовок

Рассмотрим TCP-заголовок поле за полем.

Поля *«Порт источника»* (Port Source) и *«Порт назначения»* (Port Destination) являются идентификаторами локальных конечных точек соединения. Каждый хост может сам решать, как назначать свои порты, начиная с 1024. Номер порта вместе с IP-адресом хоста образуют уникальный 48-битовый TSAP-адрес (TSAP, Transport Service Access Point — адрес на транспортном уровне). Пара номеров сокетов получателя и отправителя служит идентификатором соединения.

Поле *«Порядковый номер»* (Sequence number) идентифицирует байт в потоке данных от отправляющего TCP к принимающему TCP. Если мы представим поток байтов, текущий в одном направлении между двумя приложениями, TCP нумерует каждый байт номером последовательности. Номер последовательности представляет собой 32-битное беззнаковое число, которое переходит через 0 по достижению значения $2^{32} - 1$.

Поле *«Номер подтверждения»* (Acknowledgement Number) содержит номер сегмента с подтверждением успешного приема (который отправитель ожидает в ответ на отосланный сегмент) в последо-

вательности получаемых подтверждений. Указанный номер заносится в поле «Порядковый номер». При этом должен быть установлен контрольный бит подтверждения ACK. После того, как соединение установлено, подтверждения посылаются постоянно.

Поле *«Длина TCP-заголовка»* (DataOffset) означает размер TCP-заголовка в 32-разрядных словах. Эта информация нужна, так как поле «Параметры» может быть переменной длины, а вместе с ним и весь заголовок.

Следом идет неиспользуемое 6-битовое поле (Reserved). Это резервное поле должно быть заполнено нулями.

Затем следуют шесть 1-битовых флагов (Control Bits):

• URG — устанавливается в 1 в случае использования поля «Указатель» на срочные данные.

• ACK — установленное в 1, означает, что поле «Номер подтверждения» содержит осмысленные данные. В противном случае данный сегмент не содержит подтверждения и поле «Номер подтверждения» просто игнорируется.

• PSH — является флагом PUSH, с помощью которого отправитель просит получателя доставить данные приложению сразу по получении пакета, а не хранить его в буфере, пока буфер не наполнится, что получатель может делать ради большей эффективности.

• RST — используется для сброса состояния соединения, которое из-за сбоя хоста или по другой причине попало в тупиковую ситуацию. Кроме того, он используется для отказа от неверного сегмента или от попытки создать соединение. Если вы получили сегмент с установленным битом RST, это означает наличие какой-то проблемы.

• SYN — применяется для установки соединения. У запроса соединения бит SYN = 1, а бит ACK = 1, это означает, что поле подтверждения не используется. В ответе на этот запрос содержится подтверждение, поэтому значения этих битов в нем равны: SYN = 1, ACK = 1. Таким образом, бит SYN используется для обозначения сегментов «CONNECTION REQUEST» и «CONNECTION ACCEPTED», а бит ACK — чтобы отличать их друг от друга.

• FIN — используется для разрыва соединения. Он указывает, что у отправителя больше нет данных для передачи. Однако, даже закрыв соединение, процесс может продолжать получать данные неопределенно долго. У сегментов с битами FIN и SYN есть порядковые номера, что гарантирует правильный порядок их выполнения.

Управление потоком в протоколе TCP осуществляется при помощи скользящего окна переменного размера.

Поле *«Размер окна»* (Window) сообщает, сколько байтов может быть послано после получившего подтверждения байта. Значение по-

ля «Размер окна» может быть равно нулю; это означает, что все байты вплоть до номера подтверждения получены, но у получателя в данный момент какие-то проблемы и остальные байты он пока принять не может. Разрешение на дальнейшую передачу может быть выдано отправкой сегмента с таким же значением поля «Номер подтверждения» и ненулевым значением поля «Размер окна».

Поле *«Контрольная сумма»* (Checksum) призвано повысить надежность. Оно содержит контрольную сумму заголовка, данных и псевдозаголовка. При выполнении вычислений поле «Контрольная сумма» устанавливается на ноль, а поле данных дополняется нулевым байтом, если его длина представляет собой нечетное число. Алгоритм вычисления контрольной суммы просто складывает все 16-разрядные слова в дополнительном коде, а затем вычисляет дополнение для всей суммы. В результате, когда получатель считает контрольную сумму всего сегмента, включая поле «Контрольная сумма», результат должен быть равен 0.

Поле *«Указатель на срочные данные»* (Urgent Pointer) используется совместно с управляющим битом URG. Число, помещаемое в это поле, указывает на конец срочных данных. Срочные данные передаются вне очереди (вне потока — out of band).

Поле *«Параметры»* (Options) предоставляет дополнительные возможности, не покрываемые стандартным заголовком. Один из наиболее важных параметров позволяет каждому хосту указать максимальный размер поля полезной нагрузки, который он может принять. Использование сегментов большего размера является более эффективным, так как при этом снижается удельный вес накладных расходов в виде заголовка, однако не все хосты способны принимать очень большие сегменты. Хосты могут сообщить друг другу максимальный размер поля полезной нагрузки во время установки соединения. По умолчанию этот размер равен 536 байтам. Все хосты обязаны принимать TCP-сегменты размером $536 + 20 = 556$ байт. Для двух направлений могут быть установлены различные значения размера поля полезной нагрузки.

Если параметры занимают не полностью 32 битовое поле, то остаток заполняется нулями. Это действие называется выравниванием (Padding). Для линий с большой скоростью передачи и большой задержкой окно размером в 64 Кбайт оказывается слишком маленьким [7, 27].

Трёхэтапное установление соединения TCP. Протокол TCP обеспечивает надёжное соединение. Между двумя станциями должно быть установлено соединение TCP, прежде чем станет возможной передача данных между ними.

Соединение становится активным только после того, как отправитель и получатель обменяются несколькими управляющими пакетами для установки соединения. Данный процесс известен как *трёхшаговое рукопожатие*. Цель процедуры:

- проверить существование другого компьютера и его готовность к приёму данных;
- синхронизировать порядковые номера пакетов и номера подтверждений для каждой конечной точки в момент установки соединения TCP.

TCP является протоколом, который ориентируется на согласованную работу рабочих станций в сети и программного обеспечения партнеров, участвующих в обмене информацией.

В начале соединения каждый компьютер выбирает для первого сообщения TCP *начальный номер последовательности*. Затем с каждым новым сообщением система увеличивает этот номер на 1.

Установление связи клиент-сервер осуществляется в три этапа:

1. Клиент посылает SYN-пакет с указанием номера порта сервера, который предлагается использовать для организации канала связи.

2. Сервер откликается, посылая свой SYN-пакет содержащий идентификатор. Начальный номер ISN (Initial Sequence Number — начальный номер последовательности) не равен нулю. Процедура называется passive open.

3. Клиент отправляет подтверждение получения SYN-пакета от сервера с идентификатором, равным ISN (сервера)+1 [1].

Рассмотрим, используя рис. 5.54, типовой диалог между двумя объектами прикладного уровня с использованием протокола TCP [8]:

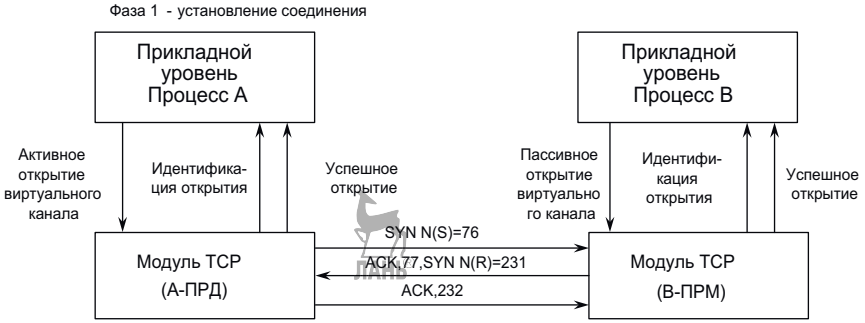
1. Для открытия виртуального соединения передатчик (модуль A) посылает флаг SYN в сегменте, с которого начнется передача (N(S) = 76).

2. Приёмник отвечает сегменту, в котором флаг ACK установлен в 1 и указывает номер байта, с которого он начнёт передавать (SYN N(R) = 231). В заголовке этого же сегмента в поле «Номер подтверждения» приёмник указывает, что он ожидает от передатчика байт с номером 77. Здесь же передаётся флаг синхронизации SYN.

3. Передатчик (модуль A), получив этот сегмент с подтверждением о готовности приёмника работать, так же отвечает сегментом с подтверждением ACK, и в поле «Номер подтверждения» передатчик указывает, что он ожидает от приёмника байт с номером 232 [6].

После этого виртуальное соединение установлено, о чем модули TCP извещают свои прикладные процессы.

На рис. 5.55 рассмотрим взаимодействие узлов в режиме «прикладной уровень — TCP».



Взаимодействие узлов в режиме «прикладной уровень — TCP»: N(S) — номер байта, с которого начнёт передавать данные передатчик (ПРД), например, 76; N(R) — номер байта, с которого будет передавать приёмник (ПРМ), например, 231.

Рис. 5.54. Диалог между двумя объектами прикладного уровня с использованием протокола TCP

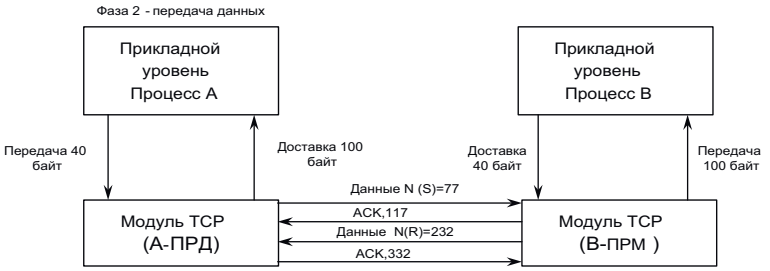


Рис. 5.55. Взаимодействие узлов в режиме «прикладной уровень — TCP»

1. С прикладного уровня передатчика (процесс А) по созданному виртуальному соединению передаются данные (40 байт), начиная с байта под номером 77.

2. Приёмник, ожидает байт данных именно с этим номером, поэтому после приёма данных приёмник выдаёт сегмент с флагом подтверждения ACK и номером следующего ожидаемого байта $N(S) = 77 + 40 = 117$. Кроме того, приёмник с прикладного уровня отсылает в сторону передатчика 100 байт данных, начиная с номера 232, что и ожидает передатчик.

3. Получив 100 байт от приёмника, передатчик выдаёт сегмент с флагом ACK и номером следующего ожидаемого байта $N(R) = 232 + 100 = 332$.

В фазе «передачи данных» работают механизмы обеспечения надёжной доставки.

3 варианта обратной связи:

– обратная связь с ожиданием;

– оконный режим переспроса;
– адресный режим переспроса.

1. Обратная связь с ожиданием — это наиболее простой вариант (рис. 5.56) [6].

На каждый передающийся сегмент ожидается получение квитанции (ACK = 1 и номер следующего запрашиваемого байта). Включается таймер ожидания. Если квитанция не приходит до истечения времени, то осуществляется повторная передача (см. рис. 5.57) [6, 8].

При потере пакета через определенный интервал времени выполняется его повторная передача.

Пунктирной линией показан процесс нормальной передачи пакета и получения подтверждения.

Причинами потери пакетов могут являться ошибки, возникающие в заголовке IP, истечение времени жизни, переполнение буфера маршрутизатора и т.п. Во всех этих случаях отправляется сообщение ICMP. Время ожидания квитанции зависит от расстояния до получа-

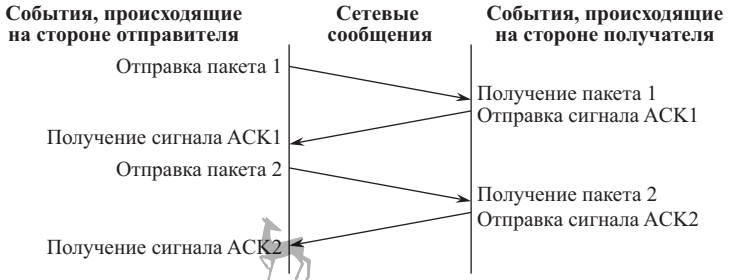


Рис. 5.56. Иллюстрация механизма подтверждения приема с повторной передачей, при котором отправитель ждет уведомления об успешном получении каждого пакета

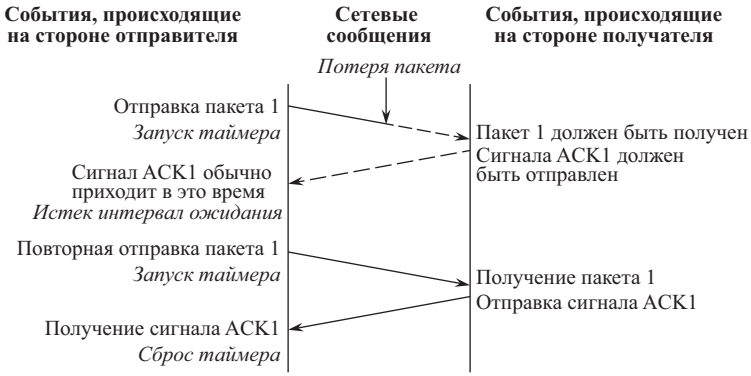


Рис. 5.57. Потеря пакета

теля (от времени двойного пробега).

Существует 2 механизма определения времени двойного пробега:

1. Тестирование сети и оценка в режиме пингования временной задержки на каждом сегменте. Затем осуществляется набор статистики и её обработка. Для определения времени ожидания квитанции к среднему времени задержки добавляются 4 среднеквадратических отклонения времени задержки от его математического ожидания. Именно за это время и должна прийти квитанция. Если квитанция приходит после заданного времени, передающая сторона считает пакет утерянным.

2. Динамический способ определения времени. Время двойного пробега определяется в процессе передачи. Первоначально передаются сегменты по одному в режиме РОС ОЖ и определяется время пробега. Время таймера увеличивается до тех пор, пока квитанции не станут успевать приходить.

К плюсам РОС ОЖ можно отнести простоту реализации. К минусам — непроизводительное использование пропускной способности и следовательно — низкая эффективная скорость ПД [6].

2. Оконный режим переспроса. Для данного режима характерно некоторое число сегментов, передаваемых непрерывно без ожидания квитанции. Оконный режим использует принцип конвейерной передачи. Таймер включается после передачи последнего сегмента (рис. 5.58) [6, 8, 28].

Рис. 5.58. Пример движущегося окна протокола TCP

Байты 1 и 2 успешно доставлены получателю; байты 3–6 посланы в сеть, но подтверждение об их доставке еще не получено; байты 7–9 еще не отправлены, но могут быть отправлены без всяких задержек; байты с номерами 10 и выше не могут быть посланы в сеть до тех пор, пока не попадут внутрь окна.

Возможны 2 режима окна: фиксированное и скользящее.

Режим фиксированного окна. Переспрос всего окна осуществляется в случае, если хотя бы 1 сегмент принят с ошибкой (РОС-НП, но для всего окна). Эффективность использования лучше, чем при РОС-НП (см. рис. 5.59) [6, 28].

Идея заключается в том, что отправитель может послать в сеть сразу все три пакета, не дожидаясь сообщений о подтверждении их приема.

Режим скользящего окна. Переспрос осуществляется с первого ошибочного сегмента. В этом режиме подтверждения, следующие за ошибочным сегментом, не передаются. Окно смещается на число правильно принятых сегментов. Этот режим наиболее эффективен.

3. Адресный режим переспроса. Конвейерная передача в размере окна. Все принимаемые сегменты накапливаются, анализируются. Определяются номера сегментов, которые приняты с ошибкой и осуществляется повторная передача только этих сегментов (рис. 5.60) [6, 8, 28, 29].

Если для какого-либо пакета, попавшего в окно, не получен сиг-

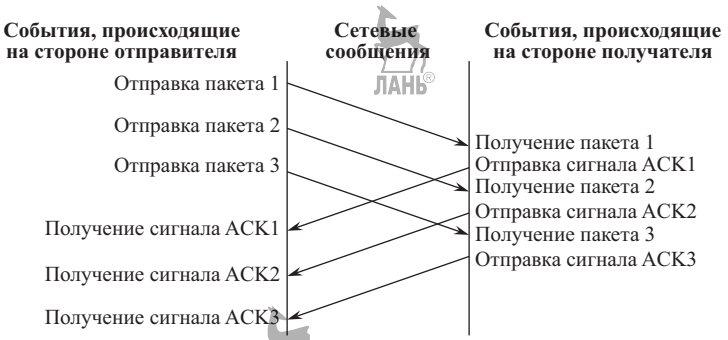


Рис. 5.59. Пересылка трех пакетов с использованием метода движущихся окон

Рис. 5.60. Движущееся окно, внутрь которого помещено 8 пакетов (а); при получении подтверждения о приеме пакета 1 окно сдвигается на один пакет вправо и в сеть отправляется 9-й пакет (б)

нал подтверждения, то выполняется повторная передача этого пакета.

Режим окна определяется во вспомогательных параметрах TCP-заголовка.

Рассмотрим плавное закрытие соединения (рис. 5.61).

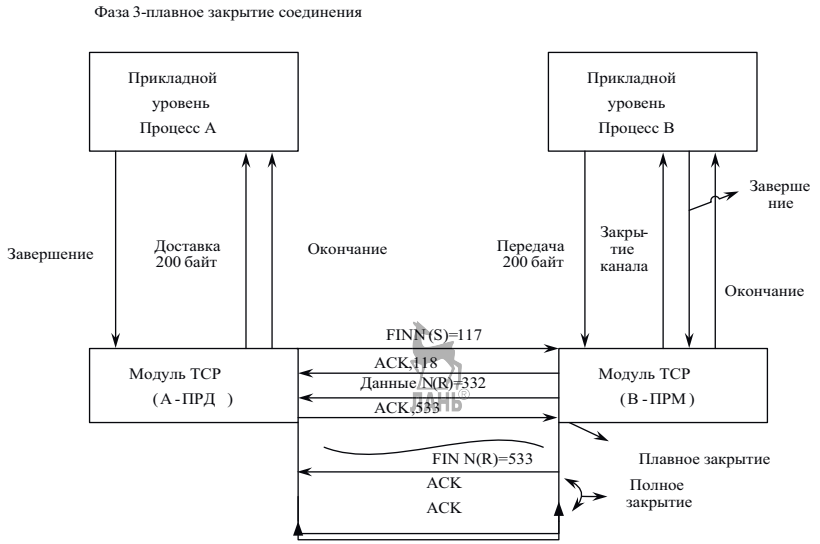


Рис. 5.61. Плановое закрытие соединения

Для плавного завершения соединения передатчик отправляет сегмент с флагом FIN и номером байта $N(S) = 117$. Приёмник выдаёт сегмент с флагом подтверждения ACK и номером ожидаемого байта $N(S) = 118$, но у приёмника ещё остались данные для передачи, которые он и отправляет (200 байт), начиная с байта под номером $N(R) = 332$. Передатчик отвечает сегментом с флагом подтверждения ACK и номером ожидаемого байта 533 $(332 + 200 + 1)$. На этом виртуальное соединение прикладного уровня разрывается, но остается ещё виртуальное соединение транспортного уровня. Для его разрушения приёмник посылает сегмент с флагом FIN и номером ожидаемого байта $N(R) = 533$. Передатчик отвечает подтверждением, на что приёмник также отвечает сегментом подтверждения ACK. На этом виртуальное соединение на транспортном уровне разрушается [6, 8, 28, 29].

5.12. Протокол UDP (User Datagram Protocol)

Назначение протокола. Протокол UDP проектировался для создания в объединенной системе компьютерных сетей с коммутацией пакетов режима передачи дейтаграмм клиента. Протокол UDP предполагает, что нижестоящим протоколом является Internet (IP).

Данный протокол предоставляет прикладной программе процедуру для посылки сообщений другим программам, причем механизм протокола минимален. Протокол UDP ориентирован на транзакции; получение дейтаграмм и защита от дублирования не гарантированы. Приложения, требующие гарантированного получения потоков данных, должны использовать протокол управления пересылкой (TCP) [30].

Протокол пользовательских дейтаграмм (UDP). В стеке протоколов TCP/IP UDP обеспечивает основной механизм, используемый прикладными программами для передачи дейтаграмм другим приложениям. UDP предоставляет протокольные порты, используемые для различения нескольких процессов, выполняющихся на одном компьютере. Помимо посылаемых данных каждое UDP-сообщение содержит номер порта-приемника и номер порта-отправителя, делая возможным для программ UDP на машине-получателе доставлять сообщение соответствующему реципиенту, а для получателя посылать ответ соответствующему отправителю.

UDP использует Internet Protocol для передачи сообщения от одной машины к другой и обеспечивает ту же самую ненадежную доставку сообщений, что и IP. UDP не использует подтверждения прихода сообщений, не упорядочивает приходящие сообщения и не обеспечивает обратной связи для управления скоростью передачи информации между машинами. Поэтому UDP-сообщения могут быть потеряны, размножены или приходить не по порядку. Кроме того, пакеты могут приходить раньше, чем получатель сможет обработать их. В общем, можно сказать, что UDP обеспечивает ненадежную службу без установления соединения и использует IP для транспортировки сообщений между машинами. Он предоставляет возможность указывать несколько мест доставки на одном компьютере.

Прикладные программы, использующие UDP, несут полную ответственность за проблемы надежности, включая потерю сообщений, дублирование, задержку, неупорядоченность или потерю связи. К несчастью, программисты часто игнорируют эти проблемы при разработке программ. Кроме того, поскольку программисты тестируют свои программы, используя надежные высокоскоростные локальные сети, тестирование может не выявить возможные ошибки. Таким образом, программы, использующие UDP и успешно работающие в локальной сети, будут аварийно завершаться в глобальных сетях TCP/IP [30].

Формат UDP-сообщений. Каждое UDP-сообщение называется пользовательской дейтаграммой. Концептуально дейтаграмма состоит из двух частей: UDP-заголовка и области данных UDP. Как показано на рис. 5.62, заголовок состоит из четырех 16-битных полей, кото-

рые определяют порт, из которого было послано сообщение, порт, в который сообщение приходит, длину сообщения и контрольную сумму UDP.

Рис. 5.62. Формат полей в дейтаграмме UDP

Поля «Порт отправителя UDP» и «Порт получателя UDP» содержат 16-битные номера портов, используемые для разделения сообщений, получения которых ожидают процессы. Поле «Порт отправителя» необязательно. Когда оно используется, оно обозначает порт-источник сообщения, на который нужно посылать ответы, если не используется, оно должно содержать ноль.

Поле «Длина сообщения UDP» содержит число байт в дейтаграмме, включая заголовок UDP и данные.

Контрольная сумма UDP необязательна, значение 0 в поле «Контрольная сумма» означает, что сумма не вычисляется. Разработчики решили сделать контрольную сумму необязательной, чтобы уменьшить объем вычислений при использовании UDP в высоконадежной локальной сети. Заметим, однако, что IP не вычисляет контрольную сумму поля данных в IP-дейтаграммах. Таким образом, контрольная сумма UDP обеспечивает единственную гарантию того, что целостность данных сохранена и ими можно пользоваться.

5.13. Контрольные вопросы

1. Сеть класса A. Необходимо организовать 64 подсети, определить маску подсетей, количество хостов в каждой подсети, адрес подсетей 12, 43, 56.
2. Сеть Интернет имеет адресацию класса B. Необходимо организовать 38 подсетей. Определить маску подсетей, диапазон адресов сети данного класса и адреса подсетей 7, 12, 26.
3. Сеть Интернет имеет адресацию класса A. Необходимо организовать 56 подсетей. Определить маску подсетей, диапазон адресов сети данного класса и адреса подсетей 6, 14, 36.
4. Сеть Интернет имеет адресацию класса C. Необходимо организовать 4 подсети. Определить маску подсетей, диапазон адресов сети данного класса и адреса всех подсетей.
5. Пусть IP-адрес узла подсети равен 198.65.12.67, а значение маски для этой подсети 255.255.255.240. Определить номер подсети. Какое максимальное число узлов может быть в этой подсети?

6. Пусть поставщик услуг Internet имеет в своем распоряжении адрес сети класса B. Для адресации узлов своей собственной сети он использует 254 адреса. Определите максимальное возможное число абонентов этого поставщика услуг, если требуемые размеры сетей, соответствуют классу C. Какая маска должна быть установлена на маршрутизаторе поставщика услуг, соединяющем его сеть с сетями абонентов?
7. Для чего используется «Метка потока» в заголовке IPv6?
8. Назначение адреса Unicast.
9. Назначение адреса Anycast.
10. Назначение адреса Multicast.
11. Назначение протокола ARP.
12. Назначение протокола RARP.
13. От чего зависит длина полей протокола ARP?
14. Что определяет в формате протокола поле «тип оборудования»?
15. Когда у протокола IP возникает необходимость обращения к протоколу ARP?
16. Назначение MAC-адреса.
17. Как создаются статические и динамические записи в ARP-таблице?
18. Какое устройство называют ARP-сервером и почему?
19. Что произойдет, если IP-адрес в ARP-таблице не будет обнаружен?
20. Что произойдет, если отправитель хочет отправить данные другому устройству и он знает IP-адрес получателя, но MAC-адрес получателя в его ARP-таблице отсутствует?
21. Какой вид имеет широковещательный MAC-адрес?
22. Что происходит, когда устройство, создавшее ARP-запрос, получает ответ?
23. Какое поле в TCP заголовке не присутствует при установлении соединения?
24. Как расшифровывается TCP?
25. Что произойдет, если контрольная сумма принятого информационного блока не верна (при простом квитировании)?
26. Когда используется бит URG (ACK, PSH, RST, SYN, FIN)?
27. Какие значения может принимать поле «Порядковый номер»?
28. Сколько необходимо блоков для установления TCP-соединения (в обычном случае)?
29. В какой спецификации описан протокол TCP?
30. Сколько необходимо блоков для разъединения TCP-соединения?

31. Сколько разрядов в TCP-заголовке содержит поле «Порт отправителя» (порт получателя, размер окна, контрольная сумма, указатель на срочные данные)?
32. Сколько разрядов в TCP-заголовке содержит поле «Порядковый номер»?
33. Сколько разрядов в TCP-заголовке содержит поле «Резерв»?
34. Сколько разрядов в TCP-заголовке содержит поле «Параметры»?
35. На каком уровне работает протокол TCP?
36. Что произойдёт, если время ожидания в TCP истечёт?
37. Как расшифровывается UDP?
38. Что произойдет, если одна из дейтаграмм UDP не достигнет места назначения?
39. В какой спецификации описан протокол UDP?
40. Из какого количества полей состоит заголовок UDP?
41. Какие поля являются не обязательными в UDP-заголовке?
42. Сколько разрядов в UDP-заголовке содержит поле «Длина UDP»?

5.14. Список литературы

1. [RFC 2453]. Kedrov, Sergey, 2000. Режим доступа: http://www.protocols.ru.
2. *Хелеби С., Мак-Ферсон Д.* Принципы маршрутизации в Internet. — 2-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2001. — 448 с.
3. *Семёнов Ю.А.* Протоколы Internet. Энциклопедия. — М.: Горячая линия–Телеком, 2001. — 744 с.
4. *Олифер В., Олифер Н.* Компьютерные сети. Принципы, технологии, протоколы. — 4-е изд. — М.: Питер, 2010. — 943 с.
5. *Столингс В.* Современные компьютерные сети. — 2-е изд. — М.: Питер, 2003. — 781 с.
6. [Электронный ресурс]. — Режим доступа: http://opds.sut.ru/old/electronic_manuals/itm_sait/tema62.htm
7. [Электронный ресурс]. — Режим доступа: http://olympic.tusur.ru/books...RIO-SEVMT...Boichenko52.pdf.
8. *Куроуз Д.Ф., Росс К.В.* Компьютерные сети. Многоуровневая архитектура Интернета. — М.: Питер, 2004. — 765 с.
9. *Кульгин М.* Технологии корпоративных сетей. Энциклопедия. — М.: Питер, 2000. — 704 с.
10. [Электронный ресурс]. — Режим доступа: http://www.intuit.ru/studies/courses/636/492/lecture/11128?page=2.
11. [Электронный ресурс]. — Режим доступа: http://www.ciscolab.ru/.

12. *Braun H.-W.* Models of Policy Based Routing, RFC 11044. — Merit / NSFNET, June 1989.
13. *Berger L., O'Malley T.* RSVP Extensions for IPSEC Data Flows, RFC 2207. — September 1997.
14. *Семенов Ю.А.* (ИТЭФ-МФТИ) http://book.itep.ru/4/44/rsv_4496.htm.
15. emanual.ru/download/9996.html.
16. [RFC1657] *Willis S., Burruss J., Chu J.* Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2, RFC 1657. — July 1994.
17. [RFC2858] *Bates T., Rekhter Y., Chandra R., Katz D.* Multiprotocol Extensions for BGP-4, RFC 2858. — June 2000. BGP протокол (перевод на русский) from http://www.rublin.org.ua/article/bgp22.
18. [RFC3392] *Chandra R. and Scudder J.* Capabilities Advertisement with BGP-4, RFC 3392. — November 2002.
19. [RFC2918] *Chen E.* Route Refresh Capability for BGP-4, RFC 2918. — September 2000.
20. [RFC-2209] Resource Reservation Protocol (RSVP) — Version 1. Message Processing.
21. [RFC 4351] Real-Time Transport Protocol (RTP) Payload for Text Conversation Interleaved in an Audio Stream.
22. [RFC 3550] RTP: A Transport Protocol for Real-Time Applications.
23. [RFC 3942] Reclassifying Dynamic Host Configuration Protocol Version Four (DHCPv4) Options.
24. *Wahl M., Howes T., Kille S.* Lightweight Directory Access Protocol (v3), IETF RFC 2251. — Dec. 1997.
25. [RFC 903] Reverse Address Resolution Protocol, Internet Standard STD 38.
26. [RFC 903] A Reverse Address Resolution Protocol, R. Finlayson, T. Mann, J. Mogul, M. Theimer (June 1984).
27. *Камер Д.* Сети TCP/IP, том 1. Принципы, протоколы и структура = Internetworking with TCP/IP, Vol. 1: Principles, Protocols and Architecture. — М.: Вильямс, 2003. — 880 с.
28. [RFC 793] Transmission Control Protocol. 1981. http://rfc.net/rfc793.html.
29. [RFC 1180] RFC Семейство протоколов TCP/IP в русском переводе. 1998. Перевод с английского: А.Ф. Брежнев, Р.Л. Смелянский. http://lib.ru/tcpbook.
30. [RFC-768] *Postel J.* RFC 768. User Datagram Protocol. 1980. http://rfc.net/768.html.

Глава 6

Транспортные IP-сети

Сеть транспортная — это совокупность ресурсов систем передачи (каналов, трактов, секций) и относящиеся к ним средства контроля, оперативного переключения, резервирования и управления, предназначенные для переноса информации между заданными пунктами сети.

До недавнего времени в качестве основного варианта транспортной IP-сети рассматривалась сеть, построенная с использованием четырехуровневого стека протоколов (рис. 6.1,а). Преимуществом такого решения является использование наиболее привлекательных особенностей каждого из четырех уровней. Так, DWDM позволяет обеспечить максимально возможную пропускную способность, SDH — возможности по защите и восстановлению при отказах, ATM-уровень обеспечивает поддержку QoS, IP-уровень — взаимодействие множества клиентов, работу с приложениями.

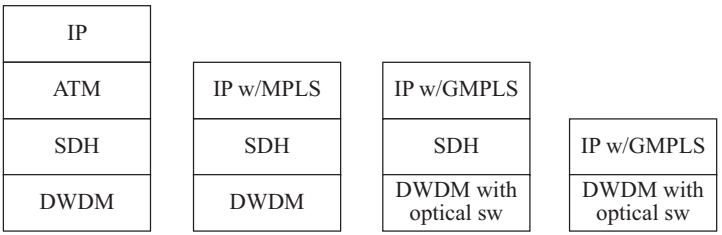


Рис. 6.1. Тенденции развития транспортных сетей

Недостатками такой четырехуровневой сети являются:

- сложность (особенно в управлении) и высокие издержки на ее создание;
- проблемы на одном из технологических уровней обязательно скажутся на всей сети;
- дублирование некоторых функций на разных уровнях.

Рассмотрим более подробно уровни ATM и SDH, представленные на рис. 6.1,а и попробуем рассмотреть пути упрощения и удешевления сети при сохранении, по крайней мере, всех ее функций (рис. 6.1).

6.1. Технология ATM

ATM (Asynchronous Transfer Mode) — метод асинхронной передачи предполагает запись любого вида информации в ячейки (Cells)

фиксированной длины. Ячейки содержат полезную информацию и заголовок. Для заголовка отводится 5 байт, для полезной информации — 48 байт.

Обнаружение и исправление ошибок осуществляется только в заголовке. Для содержимого информационных ячеек никакой проверки и восстановления не применяется, и используется передача информации, ориентированная на соединение. Реализация ATM обычно осуществляется аппаратными средствами. Все это в сочетании с статистическим мультиплексированием уменьшает время задержек, что особенно важно при передаче трафика реального времени.

Технология ATM предоставляет методы управления трафиком и механизмы качества обслуживания. Это означает, что в сетях ATM могут быть зарезервированы ресурсы, гарантирующие требуемые значения пропускной способности, задержки передачи и уровня потерь ячеек.

Стек протоколов ATM. В стеке протоколов ATM (рис. 6.2) различают следующие уровни: адаптации, ATM и физический. Уровень адаптации ATM (ATM Adaptation layer, AAL) делится на два подуровня: конвергенции (Convergence Sub-layer, CS) и сегментации и восстановления (Segmentation And Reassembly, SAR). Уровень адаптации ATM, по сути, является интерфейсом между приложениями пользователя и уровнем ATM и обеспечивает поддержку четырех различных групп (классов) приложений.

Все приложения используют один и тот же подуровень SAR, но каждый тип приложений реализует свой собственный специфический подуровень CS.

Подуровень конвергенции (CS) отвечает за получение протокольного модуля данных (Protocol Data Unit, PDU) от вышележащих уровней и их адаптацию, обычно за счет добавления служебной информации для дальнейшего представления уровню SAR. Так как каждый тип трафика требует специфической обработки, различают четыре типа уровней адаптации AAL (ATM Adaption Layer).

Задачей подуровня SAR является формирование модулей длиной 48 октетов, которые становятся полезной нагрузкой ячеек ATM.

Правило функционирования подуровня SAR заключается в том, что ничто не покидает подуровень, если его длина не равняется 48 октетам. В некоторых случаях в подуровне SAR могут добавляться свои собственные данные к модулю PDU подуровня CS, в других — он просто «нарезает» модули PDU подуровня CS на модули по 48 октетов и передает их вниз на уровень ATM.

Уровень ATM соответствует нижней части канального уровня модели OSI. Его основной задачей является коммутация ячеек способом,

Рис. 6.2. Стек протоколов ATM

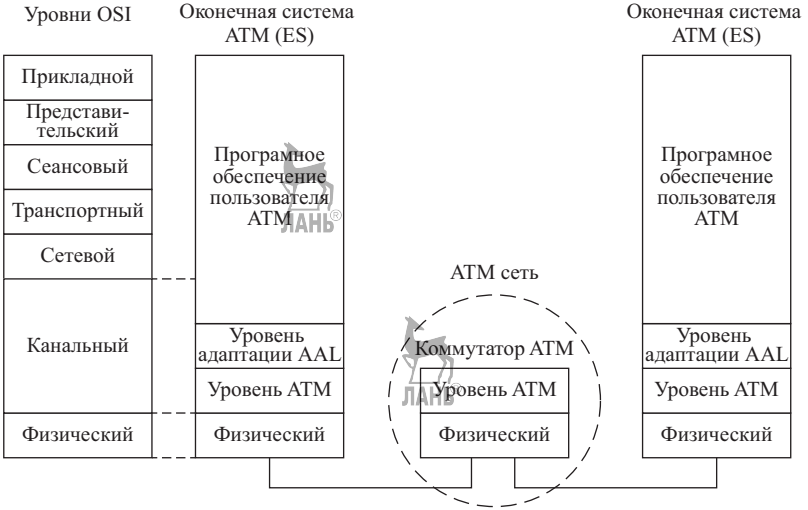


Рис. 6.3. Соответствие уровней стека протоколов ATM модели OSI

Бит	8	7	6	5	4	3	2	1	
	Управление потоком (GFC)				Идентификатор виртуального пути (VPI)				1
	Идентификатор виртуального пути (VPI)				Идентификатор виртуального канала (VCI)				2
	Идентификатор виртуального канала (VCI)								3
	Идентификатор виртуального канала (VCI)				Тип нагрузки (PT)			CLP	4
	Контрольная сумма заголовка (HEC)								5
									Байт

Рис. 6.4. Структура заголовка ячейки ATM

подходящим для осуществления их передачи между отправителем и получателем.

На рис. 6.3 представлено соответствие стека протоколов ATM модели OSI.

На рис. 6.4 изображена структура заголовка ячейки ATM. Далее будут рассмотрены только поля VPI и VCI. Определение других полей можно найти в [ITU-T I.361]. Ячейки, принадлежащие ATM-соединению, идентифицируются при помощи идентификатора ячейки (CI), который состоит из контрольных полей VPI и VCI. CI (Cell Identification) определяется при установлении соединения, и далее ячейки передаются по сети в соответствии со значением CI. VPI определяет виртуальный путь, значение VCI определяет виртуальный канал, которому принадлежат ячейки. Взаимосвязь между VPI и VCI изображена на рис. 6.5.

Соединение между двумя оконечными пунктами сети возникает тогда, когда один из них передает через интерфейс «пользователь–сеть» (UNI) (рис. 6.6) запрос в сеть. Этот запрос через цепочку ATM-коммутаторов отправляется в пункт назначения для интерпретации. Если узел-адресат принимает запрос на соединение, то в ATM-сети между двумя пунктами организуется виртуальный канал.

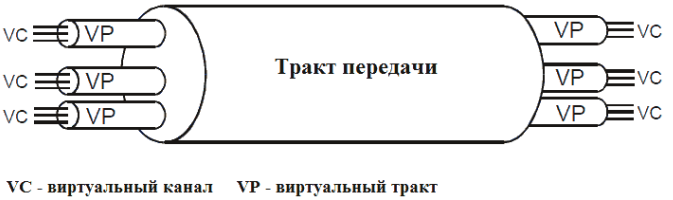


Рис. 6.5. Взаимосвязь между виртуальным каналом, виртуальным путем и трактом передачи

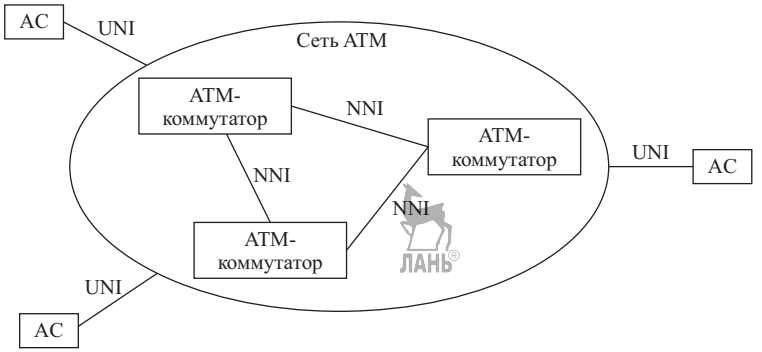


Рис. 6.6. Структура сети ATM

Принцип коммутации в ATM иллюстрируется рисунками 6.7,а и 6.7,б. На рис. 6.7,а представлен принцип коммутации путей, а на рис. 6.7,б — как коммутация путей, так и каналов.

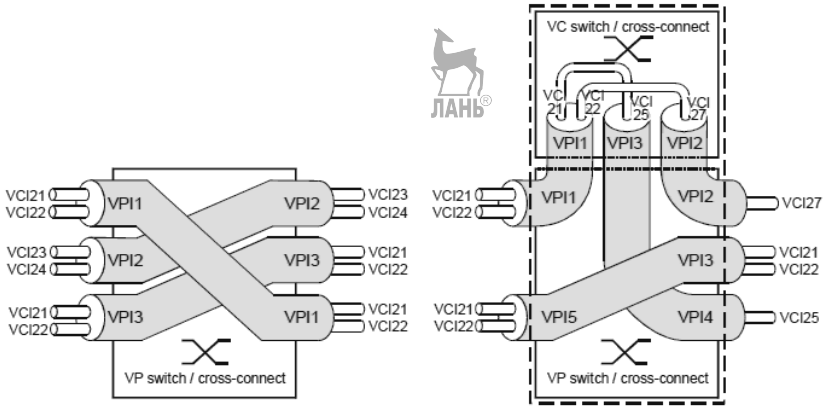


Рис. 6.7. VP и VP/VC коммутаторы/кросс-коннекторы

Заметим, что сеть ATM обеспечивает два основных типа соединений. Постоянное виртуальное соединение (Permanent Virtual Connection, PVC) — это долговременное соединение (несколько дней и даже месяцев), которое обычно устанавливается между конечным оборудованием сети ATM и используется при работе операторов. Для каждого такого соединения жестко заданы маршрут, скорость и класс обслуживания (QoS). Коммутируемое виртуальное соединение (Switched Virtual Connection, SVC) устанавливается по запросу со стороны вызывающего абонента. Соединение создается только в том случае, если имеются соответствующие ресурсы сети, и только на время, необходимое для обмена информацией. После окончания передачи пакетов или ретрансляции кадров соединение сразу разрывается.

Классы обслуживания AAL (рис. 6.8). Различают четыре класса обслуживания, охватывающие определенные типы трафика, которые, по мнению создателей ATM, встречаются в настоящее время или могут появиться в будущем.

Услуга класса A является сервисом с установлением соединения. Он поддерживает трафик с постоянной скоростью битов, который требует сквозной синхронизации. Этот класс услуг обычно используется для передачи потоковых речевых и видеосигналов без сжатия.

Услуга класса B является сервисом с установлением соединения и отличается от сервиса класса A только поддержкой сигналов с переменной скоростью передачи битов. Для трафика, который использует

сервис класса B, также требуется синхронизация. Сигналы, которым необходима услуга класса B, включают сжатые и разбитые на пакеты речевые и видеоданные.

Услуга класса C является услугой с установлением соединений и предназначена для поддержки трафика с переменной скоростью передачи данных, не требующих поддержки синхронизации. Трафик, который использует услугу класса C, может включать, но не ограничен данными, предполагающими установление соединений, такими как кадры Frame Relay.

Услуга класса D поддерживает трафик данных, ориентированный на отсутствие соединений. Такой трафик характеризуется изменчивостью скорости передачи битов и отсутствием требований к сквозной синхронизации. Примером такого трафика являются пакеты протокола IP.

Четырем типам класса обслуживания первоначально соответствовали четыре типа протоколов адаптации ATM. Впоследствии протоколы AAL3 и AAL4 были заменены протоколами AAL3/4, который оказался неэффективным. Это привело к разработке нового протокола, получившего название SEAL (Simple Efficient Adaptation Layer — простой эффективный протокол адаптации). ATM-форум после принятия этого протокола дал ему название AAL5. Будущее, по-видимому, принадлежит AAL5.

Качество обслуживания в ATM-сетях характеризуют несколько параметров: доступность пропускной способности, достоверность передачи данных, приоритет трафика, задержки в сети. Предположим, что достигается верхний предел пропускной способности сети. В такой ситуации клиент либо провайдер может выделить часть трафика как более приоритетную, и за нее пользователь будет платить от-

Трафик (класс услуги)	Звук (A)	Видео со сжатием (B)	Данные, FR, ... (C)	LAN (D)
Синхронизация	Требуется		Не требуется	
Скорость	Постоянная	Переменная	Переменная доступная	Переменная неопределенная
Соединение	Установление соединения, виртуальные каналы			Без соединения
Тип AAL	AAL1	AAL2	AAL3/4	
			AAL5	
Временной параметр	Реальное	Реальное/ нереальное	Нереальное	

Рис. 6.8. Классы обслуживания AAL

дельно. Например, если требуется уменьшить задержки для передачи видеоматериалов и голоса, клиент может заказать качество обслуживания с так называемой постоянной битовой скоростью, которое характеризуется малыми задержками в сети. Для того, чтобы сеть могла определить запрошенный уровень качества обслуживания, его значение заносят в 5-байтовый заголовок ячейки ATM.

Пользователи услуг ATM платят по более высоким тарифам за повышенные уровни качества обслуживания, такие как постоянная битовая скорость и переменная битовая скорость с поддержкой режима реального времени. С учетом своих потребностей заказчики выбирают те или иные уровни качества, чтобы оплачивать лишь то, что реально требуется. Если нужно пересылать только данные, нет смысла платить за более высокое качество, необходимое для передачи голоса в режиме реального времени.

В связи с увеличением количества приложений и услуг на основе IP технология ATM становится источником непроизводительных задержек, вследствие чего ее применение нецелесообразно при работе на высоких скоростях. С другой стороны, благодаря популяризации и совершенствованию технологий и услуг VoIP, передача голоса стала возможной на уровне IP и наряду с появившимися возможностями Traffic Engeniring в IP/MPLS исключила необходимость повсеместного развертывания ATM. ATM «мигрировала» из транспортных сетей в сети доступа в качестве канальной технологии доступа в сети xDSL.

6.2. Синхронная цифровая иерархия (SDH)

SDH (Synchronous Digital Hierarchy) была задумана как скоростная информационная автострада для транспортирования цифровых потоков с разными скоростями. В этой иерархии объединяются и разъединяются потоки со скоростями 155,520 Мбит/с и выше. Поскольку способ объединения потоков был выбран синхронный, то данная иерархия получила название синхронной цифровой иерархии [2].

Технология SDH ориентирована на соединение, в ней используется коммутация каналов, основанная на ТДМ, широко используется мультиплексирование и иерархическое деление пропускной способности тракта на малые порции. Технология предусматривает механизмы защиты от отказов и мониторинг.

Функциональная архитектура SDH. Определение функциональной архитектуры SDH дано в ITU-T G.803. В качестве физического интерфейса обычно используется оптоволокно. Также определены альтернативные физические интерфейсы для радио- и спутниковых линий передач, электрический интерфейс для низкоскоростной передачи данных. Соединения в транспортной сети SDH представлены рис. 6.9, где RS (Regenerator Section) — регенерационная секция,

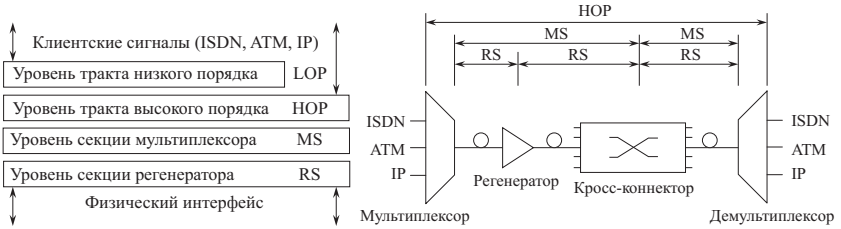


Рис. 6.9. Транспортная сеть SDH

MS (Multiplex Section) — секция мультиплексирования, HOP (High Order Path Layer) — уровень тракта высокого порядка.

Секция мультиплексирования обеспечивает передачу информации между двумя соседними сетевыми элементами, т.е. терминальными мультиплексорами (Terminal Multiplexer, TM) (Add/Drop Multiplexer, ADM).

Секция регенерации обеспечивает возможность передачи информации на протяженных участках за счет восстановления формы сигнала.

Мультиплексирование цифровых потоков в SDH. Все известные цифровые потоки в SDH транспортируются в виде цифровых информационных структур, называемых виртуальными контейнерами VC (Virtual Container, VC). Помимо блоков данных в виртуальный контейнер помещается служебная информация, в частности, заголовок пути (Path Over Head, POH) и др.

На рис. 6.10 представлена схема мультиплексирования в SDH, позволяющая получить синхронные транспортные модули STM-N, где $N = 0, 1, 4, 16, 64, 256$, отличающиеся скоростями от 51,84 Мбит/с до 39813,12 Мбит/с. В трибутарных (Tributary Unit, TU) блоках и административных блоках (Administrative Unit, AU) добавляются соответствующие указатели. Концепция указателей — ключевая в технологии SDH. Указатель определяет текущее положение виртуального контейнера в агрегированной структуре более высокого уровня, каковой является трибутарный блок, либо административный блок. Основное отличие этих блоков от виртуального контейнера заключается в наличии дополнительного поля указателя. Трибутарные блоки объединяются в группы, а те, в свою очередь, входят в административные блоки. Группа административных блоков (Administrative Unit Group, AUG) в количестве N и образует полезную нагрузку кадра STM-N [2].

На рис. 6.11 показана структура фрейма STM-N. STM-1 является базовым фреймом передачи данных с 270 столбцами и 9 строками. Столбцы первых 9 октетов зарезервированы для секционного заголовка и указателя места административной единицы внутри кад-

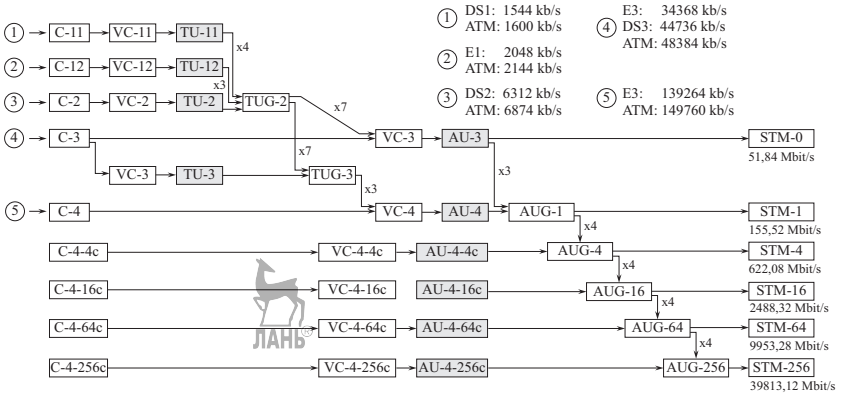


Рис. 6.10. Структура мультиплексирования SDH

Рис. 6.11. Структура фрейма SDH

ра. Секционный заголовок содержит заголовок регенеративной секции (RSOH) и заголовок секции мультиплексирования (MSOH). Для сигналов STM-*N* более высокого порядка (STM-4, STM-16, STM-256) используется побайтовое перемежение *N* фреймовых структур.

Фрейм STM-0 имеет только 90 столбцов и 9 строк. Он в основном используется для совместимости со стандартом Северо-Американской SONET (синхронной оптической сетью), который определен американским национальным институтом стандартов (ANSI). Стандарт SONET аналогичен SDH, но в нем используются другие отображения клиентских сигналов (client signal mappings) и другая структура мультиплексирования.

Элементы сети SDH. Существует четыре основных элемента сети SDH: SDH мультиплексоры, регенераторы, мультиплексоры ввода/вывода и кросс-коннекторы (рис. 6.12). SDH мультиплексор объединяет различные сигналы пользователей и сигналы нижнего уров-

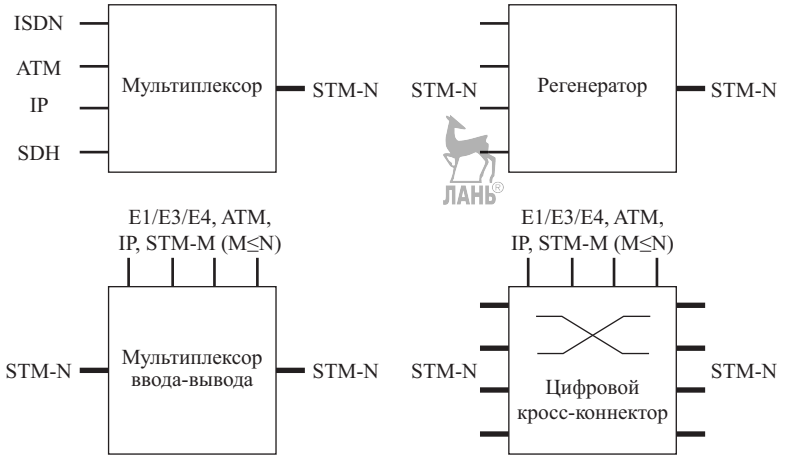


Рис. 6.12. Элементы сети SDH

ня SDH в STM-*N* сигналы. Регенератор восстанавливает сигнал, который ослабляется (затухает) в результате передачи по физической среде передачи.

Мультиплексор ввода/вывода и цифровые кросс-коннекторы обеспечивают гибкость SDH сети. В мультиплексоре ввода/вывода синхронизированные сигналы с более низкой скоростью передачи данных могут быть извлечены из фрейма STM-*N* и понижены до трибутарных портов. Новые сигналы пользователей могут быть добавлены и объединены в STM-*N* фрейм. Мультиплексоры ввода/вывода главным образом используются в сетях SDH с кольцевой топологией.

Цифровые кросс-коннекторы делают сеть более гибкой. Виртуальные канальные соединения могут быть скоммутированы (switched) между всеми (любыми) портами. В зависимости от типа оборудования коммутация может осуществляться на уровне более высокого или низкого порядка.

Сегодня технология (сети) SDH получила большое распространение. Это, помимо магистральных сетей, сети городские. Но существует ли возможность и необходимость убрать из четырехуровневого стека моделей и SDH? Понятно, что чем меньше уровней, тем проще и дешевле будет сеть в целом.

Здесь следует в первую очередь упомянуть о возрастающей роли Ethernet и возникновении понятия Carrier Ethernet Transport или E_oT. Известно, что 98% всех соединений начинаются и заканчиваются на портах Ethernet, следовательно, напрашивается вывод о целесообразности использования Ethernet и непосредственно в опорных сетях. На данный момент существуют решения, которые позволяют технологии Ethernet, изначально разработанной исключительно для локаль-

ных сетей, достичь уровня операторского класса. Эти решения обеспечат возможность заменить оборудование SDH, чему способствует существование стандартов Ethernet, предусматривающих скорости 40 и 100 Гбит/с.

Но отказ от SDH в большей степени продиктован тенденцией к полной замене TDM-трафика пакетным «All over IP-трафиком», что рано или поздно, по мнению авторов [1], устранит необходимость в SDH оборудовании.

Другие аргументы в пользу отказа от SDH — новые достижения в области защиты и восстановления связи в оптических сетях, а также последние стандарты OTN (Optical Transport Network). Существующее сегодня оборудование OADM (Optical add-drop multiplexer) и оптические кросс-коннекторы OXC позволяют достигать скоростей более 25 терабит в секунду на одно волокно и использовать исключительно оптическую коммутацию вместо традиционной схемы коммутации «оптическая–электрическая–оптическая».

Однако технология SDH сегодня все еще почти повсеместно используется в транспортных сетях. Появилась ее модификация NGSDH (Next Generation SDH) [3]. Именно NGSDH, как отмечено в [1], позволит обеспечить плавный переход к полностью оптическим сетям.

6.3. Многопротокольная коммутация по меткам

На рис. 6.1 в вариантах *б*, *в*, *г* мы видим в верхних уровнях аббревиатуры MPLS (Multiprotocol Label Switching) и GMPLS (General Multiprotocol Label Switching), что означает «многопротокольная коммутация по меткам» и «обобщенная многопротокольная коммутация по меткам».

Использование в сочетании с IP многопротокольной коммутации по меткам (IP/MPLS), как было сказано выше, позволило отказаться от ATM в опорной сети.

Ниже изложены конспективно самые минимальные сведения о MPLS и GMPLS. Поскольку профессионала от дилетанта отличает знание деталей, читайте, в случае необходимости, дополнительно материалы, которые называются RFC, посвященные MPLS и GMPLS, а также [4].

Общие положения. Слово «многопротокольная» означает, что MPLS позволяет использовать протоколы маршрутизации не только стека TCP/IP, но и любого другого стека. Главное достоинство MPLS заключается в способности предоставлять разнообразные транспортные услуги в IP-сетях и, в первую очередь, услуги виртуальных частных сетей. Виртуальные каналы MPLS обеспечивают инжиниринг трафика, так как они поддерживают детерминированные маршруты.

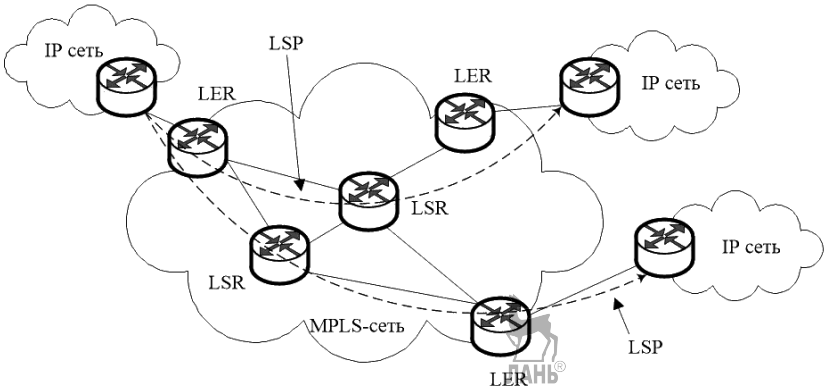


Рис. 6.13. MPLS-сеть

На рис. 6.13 представлены два типа маршрутизаторов: пограничный маршрутизатор LER (Label Edge Router) и промежуточный маршрутизатор LSR (Label Switching Router). Устройство LER является более сложным, чем LSR. LER принимает трафик от других сетей в форме стандартных IP-пакетов, добавляет к ним метки и направляет вдоль соответствующего пути к выходному LER через несколько промежуточных устройств LSR. Пути LSP (Label Switching Path) прокладываются в сети MPLS предварительно в соответствии с топологией сети. Кроме того, существует режим инжиниринга трафика, когда пути LSP прокладываются с учетом требований к резервируемой для пути пропускной способности.

Для формирования LSP и обмена сигнальной информацией используются традиционные протоколы маршрутизации с расширениями для поддержки трафик-инжиниринга (OSPF-TE и IS-IS-TE) и протоколы обмена метками — LDP (Label Distribution Protocol). Для создания LSP входной LER отсылает пакет с запросом по направлению к выходному LER. При прохождении ответа в обратном направлении каждый промежуточный LSR настраивает локальную таблицу маршрутизации для добавления соответствующей метки пакетам, принадлежащим данному потоку.

LER выполняет направление входного трафика в один из исходящих из LER путей LSP в соответствии с *классом эквивалентности продвижения* FEC (Forwarding Equivalence Class). Классом эквивалентности продвижения называется группа IP-пакетов, имеющих одни и те же требования к условиям транспортировки. Все пакеты, принадлежащие данному FEC, продвигаются через MPLS-сеть по одному виртуальному пути LSP.

В LER имеется база данных классов FEC, каждый класс опи-

сывается набором элементов, а каждый элемент описывает признаки, на основании которых входящий пакет относят к тому или иному классу [2].

Классификация FEC может выполняться:

- *на основании IP-адреса назначения*. Это наиболее близкий к принципам работы IP-сетей подход, который состоит в том, что для каждого префикса сети назначения, имеющегося в таблице LER-маршрутизации, создается отдельный класс FEC. Протокол LDP, который мы далее рассмотрим, полностью автоматизирует процесс создания классов FEC по этому способу;
- *в соответствии с требованиями инжиниринга трафика*. Классы выбираются таким образом, чтобы добиться баланса загрузки каналов сети;
- *в соответствии с требованиями VPN*. Для конкретной виртуальной частной сети клиента создается отдельный класс FEC;
- *по типам приложений*. Например, трафик IP-телефонии (RTP) составляет один класс FEC, а веб-трафик — другой;
- *по интерфейсу*, с которого получен пакет;
- *по MAC-адресу назначения кадра*, если это кадр Ethernet.

Как видно из приведенных примеров, при классификации трафика в MPLS могут использоваться признаки не только из заголовка IP-пакета, но и многие другие, включая информацию канального (MAC-адреса) и физического (интерфейс) уровней.

Концепция FEC — это, по сути дела, агрегирование потоков, т.е. объединение нескольких единичных потоков с одинаковой меткой в один с заданным качеством обслуживания. Такой подход дает возможность организовать предоставление дифференцированных услуг агрегированным потокам.

Выходное устройство LER удаляет метку и передает пакет в следующую сеть уже в стандартной форме IP-пакета.

Понятие об обобщенной многопротокольной коммутации по меткам (GMPLS). В GMPLS (General Multiprotocol Label Switching) используется универсальная метка, которая могла бы одинаково эффективно идентифицировать соединение LSP (Label Switched Path), использующие для передачи трафика различные транспортные среды.

В MPLS присвоение меток осуществляется достаточно простым способом, а именно: вышестоящий MPLS-коммутатор LSR (Label Switch Router) отправляет нижестоящему запрос на присвоения метки по одному из протоколов RSVP (RSVP PATH) или CR-LDP (CR-LDP Label request). Нижестоящий коммутатор LSR присваивает первую

свободную метку и отправляет ее значение обратно вышестоящему.

В GMPLS принцип коммутации по меткам расширен применительно к оптическим сетям. Здесь, в отличие от MPLS, вместе с меткой необходимо передавать информацию о ее типе, поскольку в качестве меток могут быть выбраны различные компоненты — длина волны λ, номер оптического волокна в канале, номер SDH-контейнера и т.д. В настоящий момент предложены следующие базовые типы меток:

- Packet — метка, идентифицирующая Ethernet (GE, FE);
- PDH — метка, идентифицирующая кадры ETSI/ANSI POH (T1, E1, E3);
- SONET/SDH — метка, идентифицирующая контейнеры SONET/SDH (VT, VC, STS-n, STM-n);
- Digital Wrapper — метка OTN G.709 (2,5, 10, 40 Гбит/с);
- λ — длина волны при использовании фотонных λ-коммутаторов OXC;
- Fiber — метка, идентифицирующая номер оптического волокна;
- Fiber Channel — метка, идентифицирующая оптический канал.

Перечисленные выше типы меток описывают тип устанавливаемого соединения LSP, а не транспортной технологии, через которую данный LSP устанавливается. Например, использование метки λ означает, что устанавливаемое соединение LSP следует обеспечивать прозрачно без оптико-электрических преобразований. Тип метки Ethernet означает, что следует также обеспечить синхронизацию и, возможно, согласование скоростей на транзитных коммутаторах.

В свою очередь, при запросе, например, метки SONET/SDH необходимо указывать тип и количество контейнеров. Таким образом, сигнальные сообщения протоколов RSVP и CP-LDP модифицированы для поддержки расширенных типов меток.

MPLS-TP [5–8]. Увеличение доли услуг на основе пакетов и увеличение объема пакетного трафика заставляют поставщиков услуг искать решения для транспорта пакетного трафика, которые были бы управляемыми, масштабируемыми и имели такой же уровень надежности, как SDH (99,999%).

Основные задачи такого рода пакетного транспорта следующие:

- установка соединений, которая происходит не часто и на длительное время;
- предоставление пользователю нескольких услуг (на основе каждого соединения);
- обеспечение высокого уровня защиты и доступности;
- гарантированное соблюдение приоритета наиболее важных услуг через механизмы QoS.

Широко внедряемая и набирающая популярность технология IP/MPLS не удовлетворяет в полной мере всем требованиям пакетного транспорта. Проблемы включают в себя отсутствие сквозного надежного управления каналами и их мониторинга, ориентация на однонаправленные соединения (а не на двустороннюю связь «точка–точка», лежащую в основе транспортных сетей). IP/MPLS — это, по сути, не ориентированная на соединения технология, так как решение о продвижении принимается в сети, а не на основе предварительного инжиниринга. Автоматическое установление и реконфигурирование соединений через сигнальные протоколы IP-маршрутизации не допускают прямого контроля потоков трафика через сеть и приводят к тому, что управление такой сетью в большей степени исправляет ошибки, чем предотвращает их. И, наконец, большая ориентация на сетевые сервисы, сложность конфигурации, сильное отличие от методологий работы с сетями SDH и дороговизна IP/MPLS-интерфейсов — это дополнительные аргументы в пользу новых технологий, ориентированных на транспорт пакетов.

Первоначально поставщики телекоммуникационного оборудования предложили облегченную версию MPLS, содержащую только ту часть MPLS, которая необходима для создания туннеля, ориентированного на установление соединения. Ставилась задача, с одной стороны, упростить IP/MPLS, а с другой — добавить необходимые для транспортных сетей функции и сделать его независимым от сигнальных протоколов IP-маршрутизации (отделить плоскости управления и продвижения данных от предоставляемых сетью сетевых служб более высоких уровней). Это предложение уже рассмотрено ITU, оно получило название T-MPLS. Эта технология должна была упростить структуру и управление по сравнению с полной версией MPLS, что обещало привести и к удешевлению оборудования. Однако проблемы совместимости нового стандарта с существующими устройствами MPLS заставило ITU вернуть в T-MPLS исходные функции MPLS. В апреле 2008 г. создается объединенная рабочая группа с IETE для проработки вопросов совместимости. Результат: в июне того же года происходит официальный отказ от T-MPLS в пользу нового MPLS-TP, т.е. MPLS транспортного профиля.

MPLS-TP основан на том же архитектурном принципе сетевых уровней, что используется сегодня в крупномасштабных сетях OTN и SDH/SONET. Операторы связи уже имеют разработанные процессы управления и высокоуровневые рабочие процедуры, основанные на этих принципах. MPLS-TP обещает стать тем решением, которое обеспечит трансформацию знакомых и надежных пакетных технологий (таких как IP/MPLS) в форму, принятую в организационных

процессах традиционных транспортных сетей с коммутацией каналов, добавив к этому возможность обслуживания Ethernet и других клиентно-ориентированных сервисов.

MPLS-TP — это ориентированная на соединения пакетная сеть на основе MPLS, которая обеспечивает управляемые сквозные соединения к сетям клиентского уровня (таким как Ethernet). Это специально выделенная реализация MPLS, где удалены все лишние функции, не имеющие отношения к коммутации пакетных соединений, и добавлены ключевые функциональные возможности, такие как QoS, сквозной OAM и зарезервированная коммутация, обеспечив тем самым полную детерминированность сети.

Таким образом, MPLS-TP — это новая разновидность MPLS, специально предназначенная для применения в транспортных сетях. Она опирается на хорошо известные и широко использующиеся технологии и стандарты сетей IP/MPLS, но без всей его избыточности, не имеющей отношения к приложениям на основе соединений, и без пробелов в транспортной функциональности, MPLS-TP может рассматриваться как основа сетей Ethernet и транспортных сетей OTN.

В отличие от классического MPLS, MPLS-TP не поддерживает режим без установления соединения, у него более простые возможности, он менее сложен и более управляем. Он открывает путь к транспортной технологии с низкой стоимостью коммутации на втором уровне, где устранена вся избыточность маршрутизации третьего уровня.

Подобно существующим транспортным сетям, MPLS-TP определяет сеть как совокупность взаимодействующих слоев, имеющих строгое разделение на слой клиента и слой сервера (сервиса). В соответствии с требованием IETF, любой вновь определяемый компонент протокола или новый функционал будут применяться в MPLS с учетом и совместно с MPLS-TP. Поэтому все новые компоненты и возможности будут появляться как часть инструментария MPLS, в состав которого входит расширенное подмножество MPLS-TP.

Основные компоненты MPLS-TP представлены на рис. 6.14. Здесь NMS (Network Management System) — система, в составе которой используется оборудование и программы, используемые для мониторинга, управления и администрирования сети передачи данных; PE (Provider Edge Router) — граничный маршрутизатор провайдера; e2e (end-to-end) означает из конца в конец.

Отметим также, что MPLS-TP имеет режимы работы со статической конфигурацией соединений либо динамической, используя GMPLS, MPLS-TP позволяет резервировать полосу пропускания для соединений, поддерживает QoS и имеет встроенные механизмы обес-

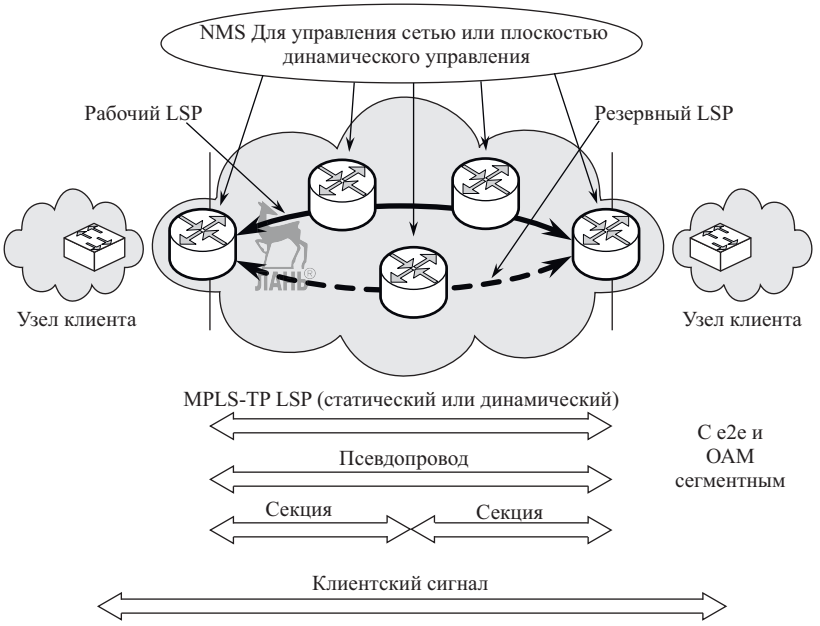


Рис. 6.14. Основные компоненты MPLS-TP

печения высокой готовности (High Availability, HA).

На рис. 6.14 представлено два типа путей с коммутацией по меткам: транспортный туннель и псевдопровод (LSP в виде виртуального канала). Метки туннелей определяет лишь входной маршрутизатор PE, с которого поступил кадр. Метки виртуальных каналов накладываются на туннели и служат в качестве демультиплексоров и указывают, к какому VPN относится кадр.

Требования к сквозной эмуляции псевдо-провода (PWE3) изложены в RFC 3916, где дается толкование понятия PWE3 (Pseudo Wire Edge to Edge). «Псевдо-провод (PW) представляет собой соединение между устройствами провайдерского края (PE), к которым подключены два AC (Attachment Circuit) — соединительных устройства». Это могут быть как физические, так и виртуальные устройства, обеспечивающие подключение CE (customer Edge), т.е. устройства, на котором сервис начинается или завершается. В качестве AC может выступать Frame Relay DLCI, ATM VPI/VCI, порт Ethernet, VLAN, канал HDLC, соединение PPP на физическом интерфейсе и т.п.

6.4. Оптическая транспортная иерархия

В обозримом будущем транспорт станет исключительно пакетным с оптической коммутацией [9–11].

Оптическая транспортная сеть OTN (Optical Transport Network)

или OTH (Optical Transport Hierarchy) строится в соответствии с рекомендациями МСЭ-Т G.709, G.798, G.872, G.873.1 и др.

Модель оптической транспортной сети представляется двумя самостоятельными по своей организации уровнями: уровнем сети OTN и уровнем пользователя.

Уровень сети OTN состоит из трех физически и логически связанных подуровней (рис. 6.15): среды передачи сигналов с разделением по длине волны WDM; оптических секций ретрансляции (Optical Transmission Section, OTS) и мультиплексирования (Optical Multiplex Section, OMS); оптических каналов (Optical Channel, OCh) с нагрузкой в виде цифровых оптических транспортных блоков (Optical Transport Unit k, OTUk) с включением в них блоков данных оптических каналов (Optical Data Unit k, ODUk), которые, в свою очередь, включают блоки полезной нагрузки оптических каналов (Optical Channel Payload Unit k, OPUk). В блоки OPU непосредственно загружаются потоки информационной нагрузки. Индекс k соответствует иерархической ступени OTH ($k = 1, 2, 3, 4$) и указывает на различные по длительности и скорости передачи циклы.

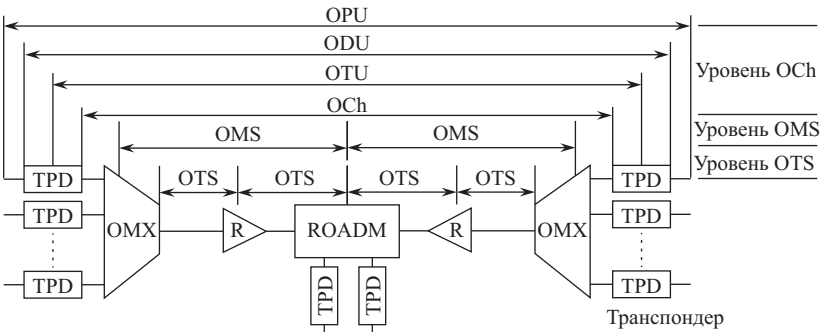


Рис. 6.15. Соединения в транспортной сети OTN

Оптические секции базируются на ресурсах одномодовых волоконных световодов со стандартными характеристиками и полосой частот передачи от 30 до 60 ТГц в диапазоне волн 1260...1625 нм. Этот диапазон используется в режиме DWDM. При этом число волновых каналов может быть реализовано от 2–4 OCh до нескольких сотен OCh, объединяемых оптическими мультиплексорами (Optical Multiplex, OMX) в оптические волновые (транспортные) модули (Optical Transport Module, OTM) емкостью до 16 OCh в каждом. Таким образом, среда передачи в этой модели транспортной сети позволяет обеспечить скорости передачи порядка 10 Тбит/с и более при скорости передачи в каждом из волновых каналов от 2,7 Гбит/с (OTU1) до 120 Гбит/с (OTU4).

Оптические секции ретрансляции OTS организуются внутри оптической секции мультиплексирования OMS для компенсации потерь оптической мощности в стекловолокне и компенсации дисперсионных искажений. Эти функции обеспечивают линейные оптические приемные волоконные усилители с эквалайзерами (обозначено на рис. 6.15 буквой R), рамановские оптические усилители и компенсаторы хроматической и поляризационной дисперсии, а в перспективе полностью оптические регенераторы 2R и 3R и волновые конверторы.

В оптической секции мультиплексирования формируются, передаются, обслуживаются и расформировываются отдельные оптические каналы, оптические волновые модули OTM с числом каналов до 16, группы оптических модулей. Каждый оптический модуль может иметь отдельный оптический сервисный канал, в который включаются служебные данные для каждого OCh. Кроме того, в секции оптического мультиплексирования создается сервисный оптический канал для обслуживания всей секции и отдельных участков — секций ретрансляции OTS. Секция OMS может иметь гарантированную защиту благодаря дублированию передачи в альтернативной кабельной линии с соответствующими секциями ретрансляции. Нормированное время защитного переключения составляет 50 мс.

Оптическая секция мультиплексирования может образовываться не только между терминальными оптическими мультиплексорами, но и между терминальными мультиплексорами и мультиплексорами ввода/вывода оптических каналов и модулей, обозначаемых ROADM (Reconfigurable Optical Add-Drop Multiplexer), т.е. изменяемых по своей конфигурации мультиплексоров ввода/вывода. Мультиплексоры ROADM позволяют вывести отдельные оптические каналы, произвести переключение этих каналов, поддержать защитную коммутацию в случае повреждений секций и т.д. Оптическая секция мультиплексирования может иметь в своем составе также оптические кросс-коннекторы. Принцип работы OADM и OXC иллюстрируется рис. 6.16 и 6.17.

Оптический канал (OCh) в оптической сети образуется транспондерными блоками (TPD) и выполняет следующие функции: преобразования электрических сигналов в оптические на передаче и оптических сигналов в электрические на приеме; регенерации цифрового сигнала типа 3R, т.е. восстанавливает амплитуду импульсов (1R), их форму (2R) и устраняет накопленные фазовые дрожания (3R) (рис. 6.18). Также производится контроль качества передачи цифровых данных в блоках OTU*k* и ODU*k*, исправление ошибок на основе избыточного кода Рида–Соломона и т.д.

Уровень оптической сети OTN может поддерживать полностью

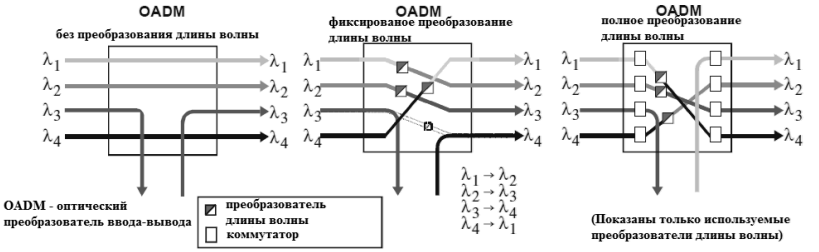


Рис. 6.16. Оптический мультиплексор ввода-вывода

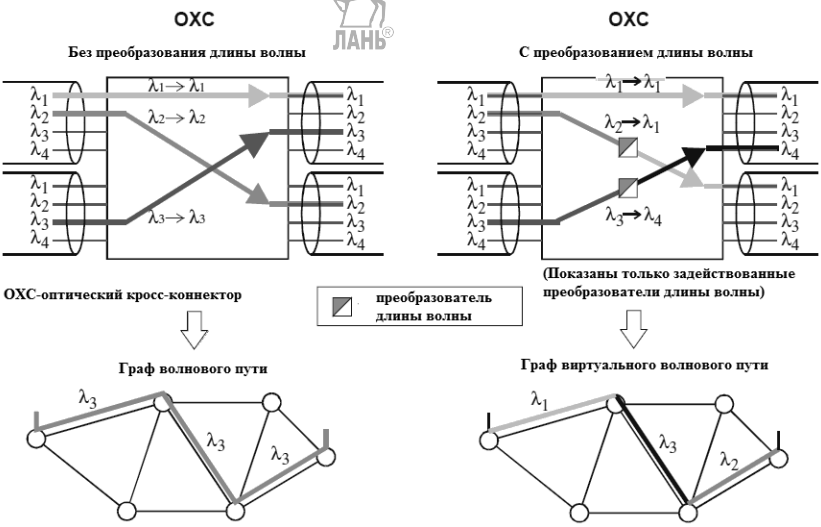


Рис. 6.17. Оптические кросс-коннекторы без и с полным преобразованием длины волны

оптическую сеть с оптической коммутацией, маршрутизацией, конвертацией оптических волн и защитой соединений.

Уровень пользователя оптической транспортной сети OTN–OTH выполняет функции интерфейса между транспортной сетью и сетями пользователей транспортных услуг, к которым относятся сети SDH, ATM, Ethernet и др. Для эффективного согласования между сетями применяются различные протокольные решения по размещению данных пользователей в оптических каналах.

Это протоколы общей процедуры формирования кадра (Generic Framing Procedure, GFP), протокол защищаемого пакетного кольца или пакетного кольца с самовосстановлением (Resilient Packet Ring, RPR) и др. Протоколы позволяют согласовать циклическую передачу данных в оптических каналах со случайной во времени передачей ин-

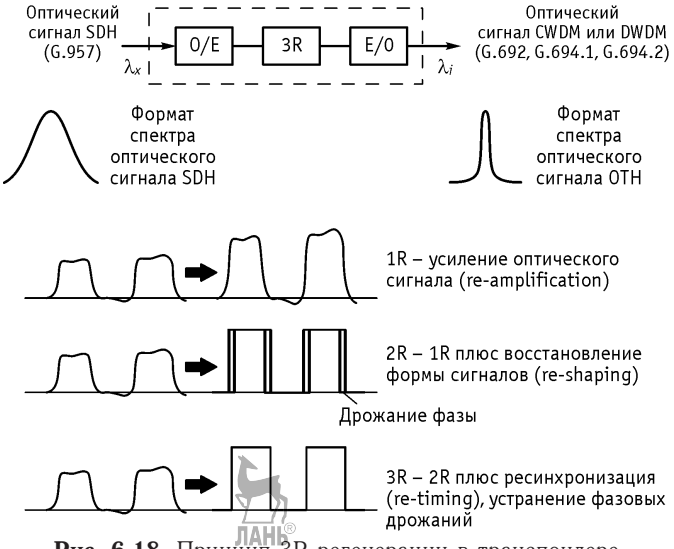


Рис. 6.18. Принцип 3R регенерации в транспондере

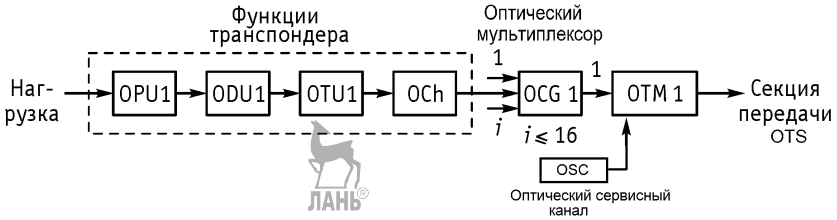


Рис. 6.19. Последовательность преобразования в OTH

формационных пакетов данных различной емкости от пользователей, например, пакетов IP, MPLS или Ethernet.

Пример цепочки цифровых и оптических преобразований представлен на рис. 6.19.

Цифровая часть схемы переходит в оптическую в блоке OCh, где электрические импульсы управляют интенсивностью оптического излучения, создаваемого лазером на вполне определенной длине волны. Таких различных волн в оптическом транспортном модуле OTM1 может объединяться в оптическом мультиплексоре, т.е. в группирователе оптических несущих (Optical Carrier Group, OCG1), до 16. К этим волнам добавляется еще одна волна оптического сервисного канала (Optical Supervisory Channel, OSC), переносящая служебные цифровые потоки для функций OAM в секциях передачи OTS, мультиплексирования OMS и в оптических каналах OCh.

6.5. Модель и иерархия Ethernet для транспортных сетей

Технология Ethernet в своем развитии прошла самый протяженный временной путь (с 1973 г. и по сей день) в сравнении со всеми известными технологиями пакетной передачи для широкополосных мультисервисных транспортных сетей.

Все этапы ее совершенствования отразились на структурах кадров, скоростных режимах и связаны с использованием, согласно модели транспортировки, на физическом уровне волоконной оптики, а на канальном уровне — быстродействующих коммутаторов, обеспечивающих формирование и управление логическим каналом (Logical Link Control, LLC) и управления доступом к среде передачи (Media Access Control, MAC). Все этапы совершенствования закреплены в стандартах IEEE (Institute of Electrical and Electronics Engineers), МСЭ-Т и MEF (Metro Ethernet Forum).

В существующих операторских сетях предлагаемые услуги были очень тесно привязаны к обеспечивающей их инфраструктуре, т.е. выделенным линиям с временным разделением (Time Division Multiplexing, TDM), которые создавались в сети PDH или SDH. Однако в первом десятилетии XXI века наметилась тенденция понимания, что основой канального уровня в новой пакетной транспортной инфраструктуре будет технология Ethernet, которая хорошо приспособлена для переноса IP-трафика, являющегося базой коммуникационных служб сетей нового поколения (Next Generation Networks, NGN). Более того, в региональных сетях пользуются спросом услуги соединения пользовательских сайтов на уровне Ethernet. Международная организация MEF описала и стандартизировала эти услуги как операторские службы Ethernet E-Line, E-LAN и E-Tree (рис. 6.20).

Однако, чтобы транспорт этих служб отвечал требованиям операторов, предъявляемым к транспортной сети, необходимо применение новых решений по Ethernet нового поколения, доведенного до уровня технологии операторского класса. К ним относится создание механизма обеспечения качества услуг (Quality of Service, QoS) и управления полосой пропускания для пользователя.

Принцип передачи кадров с сообщениями между пользовательскими терминалами демонстрируется на рис. 6.21. Мост/коммутатор (Bridge/Switch) имеет таблицу маршрутизации с адресами управления доступом к среде (MAC) и номерами портов (Port) подключения компьютеров, используя которую он пересылает кадры по соответствующим адресам терминалов-получателей DA. Терминал-получатель сообщения может также отправить сообщение в обратную сторону. Так может поддерживаться «диалог» между двумя терминалами сети. Кроме того, с одного терминала возможна посылка со-

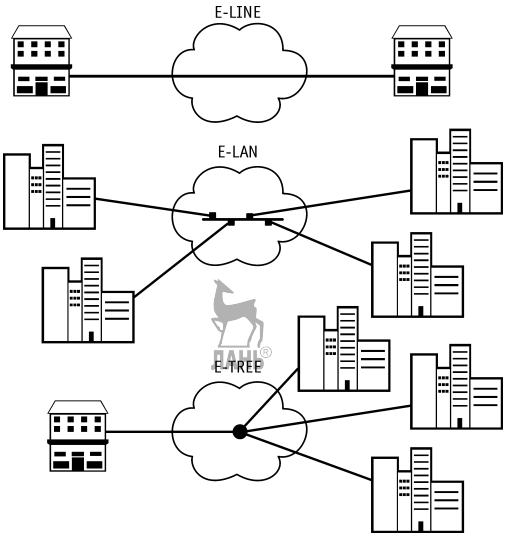


Рис. 6.20. Виртуальные соединения Ethernet по определению MEF

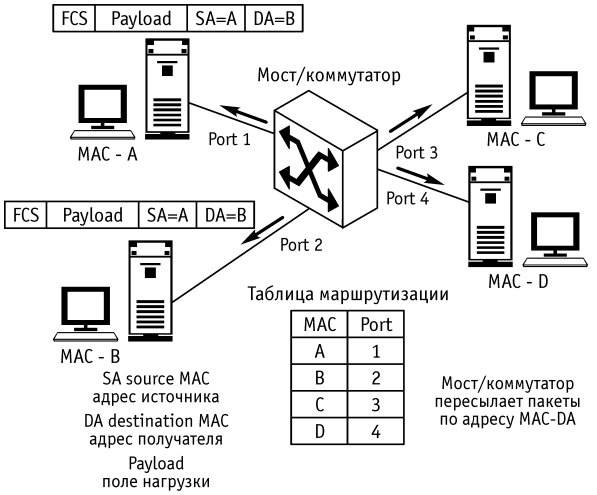


Рис. 6.21. Принцип передачи пакетов в локальной сети Ethernet

общений всем терминалам сети, т.е. групповая рассылка сообщений, что также прописывается в возможностях коммутатора.

Стандарт IEEE 802.1Q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о виртуальной локальной сети (Virtual Local Area Network, VLAN) по сети, что можно рассматривать как решение по доступу различных пользователей к различным услугам. Стандарт IEEE 802.1p специфицирует метод ука-

зания приоритета кадра, основанный на использовании новых полей, определенных в стандарте IEEE 802.1Q. К «классическому» или базовому кадру Ethernet добавлены два блока по два байта.

Необходимо отметить, что добавление четырех байтов к максимальному размеру кадра Ethernet ведет к возникновению проблем в работе многих коммутаторов, обрабатывающих кадры Ethernet аппаратно. Чтобы избежать их, группы по стандартизации предложили сократить на четыре байта максимальный размер полезной нагрузки в кадре. Для согласования работы устройств, поддерживающих формат кадра 802.1Q, с теми устройствами, которые не понимают этот формат, разработчики стандарта предложили делить весь трафик в сети на несколько типов.

Трафик входного порта (Ingress Port). Каждый кадр, достигающий коммутируемой сети и идущий либо от маршрутизатора, либо от рабочей станции, имеет определенный порт-источник. На основании его номера коммутатор должен «принять решение» о приеме (или отбрасывании) кадра и передаче его в ту или иную VLAN. Решение «судьбы» кадра, осуществляемое в единственной логической точке сети, делает возможным сосуществование самых разных видов VLAN. Приняв кадр, коммутатор «прикрепляет» к нему «ярлык» или метку (C-Tag) VLAN. Как только кадр с «ярлыком» VLAN оказывается в сети, он становится частью проходящего (Progress) или внутреннего трафика.

Внутренний трафик (Progress Traffic). Кадр с «ярлыком» коммутируется точно так же, как и без «ярлыка». Решения о его принадлежности к той или иной VLAN принимаются в пограничных элементах сети и остальные сетевые устройства индифферентно «относятся» к тому, как именно кадр попал в сеть. Так как максимальный размер кадра Ethernet остался неизменным, то пакеты всех VLAN смогут обрабатываться традиционными коммутаторами и маршрутизаторами внутренней части сети.

Трафик выходного порта (Egress Port). Чтобы попасть в межсетевой маршрутизатор или в оконечную рабочую станцию, кадр должен выйти за пределы коммутируемой сети. Ее выходное устройство «решает», какому порту (или портам) нужно передать пакет и есть ли необходимость удалять из него служебную информацию, предусмотренную стандартом 802.1Q. Дело в том, что традиционные рабочие станции не всегда воспринимают информацию о VLAN по стандарту 802.1Q, но сервер, обслуживающий несколько подсетей с помощью единственного интерфейса, должен ее активно использовать. Условное деление трафика на внутренний трафик, трафик входного и выходного портов позволяет поставщикам нестандартных реализаций

VLAN создавать шлюзы для их стыковки с VLAN, соответствующие стандарту 802.1Q. Пример разделения физической архитектуры сети Ethernet на виртуальные локальные сети отдельных групп пользователей (VLAN1, VLAN2, VLAN3) представлен на рис. 6.22.

Передача кадров производится между терминалами, соединенными общим идентификатором. В кадрах могут передаваться данные пользователей и служебная информация, относящаяся к обслуживанию сети. Передача кадров может производиться в одном из четырех режимов:

- Unicast, т.е. одноадресная передача «точка–точка». Режим определяется по MAC-адресу точки назначения;
- Multicast, т.е. многоадресная рассылка «точка – много точек». Режим задается в адресе MAC для диапазона от 01-00-5E-00-00-00 до 01-00-5E-7F-FF-FF;
- Broadcast, т.е. широковещательный режим, задаваемый адресом MAC: FF-FF-FF-FF-FF-FF;
- четвертый режим предназначен для обмена информацией обслуживания в VLAN. При этом служебные кадры могут адресоваться как определенным терминалам, так и всем, задействованным в сети.

В сети VLAN поддерживается мультиплексирование услуг нескольких виртуальных сетей, т.е. имеется возможность получать услуги от разных физических серверов, в том числе от разных провайдеров. Разработка стандарта IEEE 802.1ad Q in Q Provider Bridge (PB) позволила разделить пользователей сетей VLAN еще и на пользователей различными услугами от различных провайдеров.

Для обеспечения передачи информационных потоков в магистральных сетях связи разработана структура кадров IEEE802.1ah,

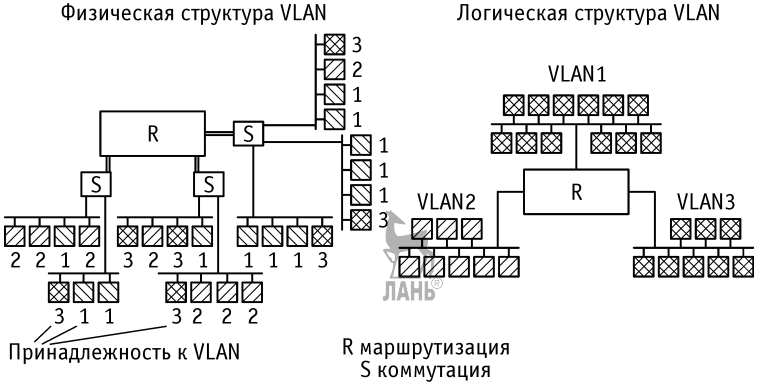


Рис. 6.22. Физическая и логическая структура с объединением пользователей в виртуальные локальные сети VLAN

в которой все метки принадлежат только транспортной сети и не зависят от транслируемых кадров IEEE802.1ad, что позволяет создавать своеобразные тоннели для переноса информационных потоков большой емкости.

Новая технология, называемая транспортной магистралью провайдера (Provider Backbone Transport, PBT), позволяет изменить рамки использования стандарта Ethernet, применение которого обычно было ограничено локальными сетями малого масштаба, и превратить Ethernet в более надежную, масштабируемую и детерминированную технологию, приспособленную для развертывания фиксированных и мобильных сетей операторского класса, что позволит предоставлять услуги видеотрансляции и видеоконференцсвязи, мультимедийные услуги, а также услуги широкополосной передачи данных и голосовой связи.

Одним из вариантов PBT является PBB-TE (Provider Backbone Bridge Traffic Engineering — мост операторских сетей с регулированием трафика), которая изначально была предложена в 2006 г. компанией Nortel под названием PBT (Provider Backbone Transport — операторский транспорт в опорной сети). Стандарт PBB-TE разрабатывался под руководством рабочей группы IEEE 802.1Qay.

PBB-TE основывается на IEEE 802.1ah PBB (Provider Backbone Bridge — операторский мост опорной сети), известный как MAC-in-MAC, и имеет с ним одинаковый формат кадра, инкапсулирующий IEEE 802.1ad (Q-in-Q) пакет с уникальными в опорной сети адресами Backbone MAC (B-MAC).

PBB-TE отказывается от присущих обычному (классическому) Ethernet функций широковещательной рассылки, обучения MAC и устранения петель (через STP).

Вместо продвижения пакетов на основе автоматических механизмов Ethernet, PBT предусматривает конфигурацию прямых соединений явным образом. Создание рабочих и резервных маршрутов переносится на уровень управления сетью.

6.6. Контрольные вопросы

1. Почему ATM-технология считается по сравнению с другими наиболее «мультисервисной»?
2. Какие функции выполняет подуровень конвергенции ATM?
3. Перечислите и дайте краткую характеристику четырем типам классов обслуживания ATM.
4. Технология SDH и ее место в транспортных сетях.
5. Из каких элементов состоит сеть с MPLS?
6. Какие задачи можно решать, используя технологию MPLS?

7. Дайте определение термина «класс эквивалентного обслуживания».
8. Что такое MPLS-TP?
9. Основные компоненты MPLS-TP.
10. Структура оптической транспортной сети.
11. Какова роль Ethernet в построении транспортных сетей?

6.7. Список литературы

1. *Ацтик А., Гольдштейн А.* Эволюция транспортных технологий // Connect. — 2009. — № 11.
2. *Олифер В., Олифер Н.* Компьютерные сети. Принципы, технологии, протоколы. — 4-е изд. — СПб.: Питер, 2010.
3. *Бакланов И.Г.* SDH > NGSDH: практический взгляд на развитие транспортных сетей — М.: Метротек, 2006.
4. *Гольдштейн А.Б., Гольдштейн Б.С.* Технология и протоколы MPLS. — СПб.: БХВ-Санкт-Петербург, 2005.
5. *Stan Hubbard* MPLS-TP in Next-Generation Transport Networks. White Paper. [Электронный ресурс]. — Режим доступа: http://fr.slideshare.net/biz4rro/hr-mplstpwp.
6. *Горнак А.* Пакетный транспорт: MPLS-TP или PBB-TP // Широкополосные мультисервисные сети. — 2009.
7. *Воробьев В.* Тенденции эволюции транспортных сетей. [Электронный ресурс]. — Режим доступа: http://nag.ru/articles/article/20069/tendentsii-evolyutsii-transportnyih-setey.html
8. *Коган С.С.* Пакетные транспортные сети: инновационные решения компании Alcatel-Lucent. [Электронный ресурс]. — Режим доступа: http://infocom.uz/2009/03/16/paketnyie-opticheskie-transportnyie-seti-innovatsionnyie-resheniya-kompanii-alcatel-lucent/
9. *Фокин В.Г.* Оптические системы передачи и транспортные сети : Учеб. пособие. — М.: Эко-Трендз, 2008. — 288 с.
10. *Фокин В.Г.* Проектирование оптической мультисервисной транспортной сети: Учеб. пособие. — Новосибирск: СибГУТИ, 2009. — 205 с.
11. *Фокин В.Г.* Оптические мультиплексоры OADM/ROADM и коммутаторы PXC в мультисервисной транспортной сети: Учеб. пособие. — Новосибирск: СибГУТИ, 2011. — 204 с.

Глава 7

Беспроводные технологии высокоскоростной передачи данных

7.1. Технология Wi-Fi (Wireless Fidelity)

Wi-Fi — название торговой марки Wi-Fi Alliance для беспроводных сетей на базе стандарта IEEE 802.11. Первоначально технология была разработана для создания беспроводных сетей LAN, но сейчас она все чаще используется как технология доступа пользователей в Интернет. Сегодня технология Wi-Fi эффективно применяется для создания MAN, RAN и даже WAN [3].

Любое оборудование, соответствующее стандарту 802.11, может быть протестировано в Wi-Fi Alliance и получить соответствующий сертификат и право нанесения логотипа Wi-Fi.

В настоящее время развивается целое семейство стандартов передачи цифровых потоков данных по радиоканалам. Параметры радиоинтерфейса, систем Wi-Fi предложены в стандартах WLAN IEEE 802.11a /b /g /h /j /n /ac и т.д. В этих стандартах предусмотрено два диапазона работы систем Wi-Fi: 2,4...2,5 ГГц (для IEEE 802.11 в/g) и 4,9...5,9 ГГц (остальные стандарты).

Скорость обмена данными в системе Wi-Fi может быть в зависимости от стандарта от 1 Мбит/с до 6 Гбит/с (для стандарта IEEE 802.11 ac) [6]. В системах может применяться как симметричный обмен данными, так и асимметричный. На физическом уровне стандарта IEEE 802.11 предусмотрены различные типы модуляции и, как следствие, различные форматы физических кадров:

- Infrared на диапазоне инфракрасных волн 850..950 нм (скорость 1–2 Мбит/с);
- FHSS на 2,4 ГГц (скорость 1 и 2 Мбит/с);
- DSSS на 2,4 ГГц (скорость 1 и 2 Мбит/с);
- HR-DSSS (скорость 1, 2, 5,5 и 11 Мбит/с);
- OFDM (скорость 6, 9, 12, 18, 24, 36, 48 и 54 Мбит/с);
- 256 QAM (скорость > 6 Гбит/с).

Сегодня в стандартах используется только DSSS, HR-DSSS, OFDM и QAM.

Современная сетевая технология коммерческого Wi-Fi предполагает разделение сети на две подсистемы: подсистемы сбора трафика в виде «пятен» (hotspot), размещаемых в точках, где можно «встретить» пользователей, и подсистемы управления, где содержится сервер идентификации пользователей, сервер биллинга, Web-портал управления сервисом, сервер авторизации Radius и пр.

Hotspot соединяются между собой через маршрутизаторы или через Интернет. Каждый пользователь, прежде чем получит доступ к коммерческой услуге, должен пройти процедуру авторизации, связанную с биллингом. Для этого он получает доступ к Web-порталу идентификации, где регистрирует свои права. После этого он может воспользоваться всеми услугами широкополосного доступа, пока на его счету остаются средства.

Внутри hotspot система может иметь внутренние коммутаторы и отдельный контроллер доступа, который выполняет роль интерфейса между всеми пользователями hotspot и распределенной сетью Wi-Fi (рис. 7.1), это особенно касается неоднородных зон покрытия сетью Wi-Fi (аэропорты, отели, бизнес-центры и пр.). Информация об услугах и их состоянии доступна на портале hotspot в рамках всего Web-портала сети [3].

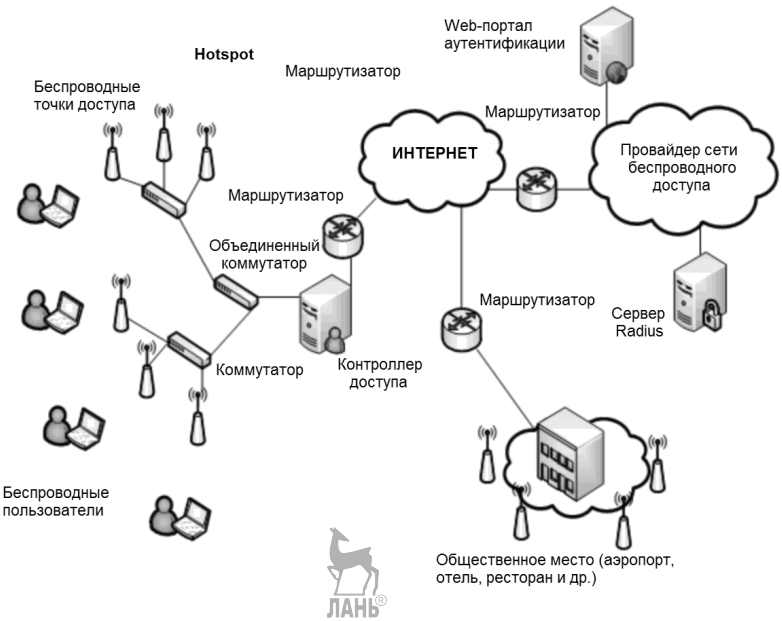


Рис. 7.1. Структура сети Wi-Fi

Технология Wi-Fi обеспечивает быстрый эффект за счет того, что одна базовая станция системы Wi-Fi предоставляет широкополосный доступ десяткам абонентов сразу, причем с довольно большими скоростями. Однако, как и любая технология, Wi-Fi обладает своими достоинствами и недостатками.

Преимущества Wi-Fi [6]:

- позволяет развернуть сеть без прокладки кабеля, что может уменьшить стоимость развертывания и/или расширения сети. Места, где нельзя проложить кабель, например, вне помещений и в зданиях, имеющих историческую ценность, могут обслуживаться беспроводными сетями;
- позволяет иметь доступ к сети мобильным устройствам;
- Wi-Fi устройства широко распространены на рынке. Гарантируется совместимость оборудования благодаря обязательной сертификации оборудования с логотипом Wi-Fi;
- мобильность. Абонент не привязан к одному месту и может пользоваться Интернетом в комфортной обстановке;
- в пределах Wi-Fi зоны в сеть Интернет могут выходить несколько пользователей с компьютеров, ноутбуков, телефонов и т.д.;
- излучение от Wi-Fi устройств в момент передачи данных на порядок (в 10 раз) меньше, чем от сотового телефона.

Недостатки Wi-Fi:

- в диапазоне 2,4 ГГц работает множество устройств, таких как устройства, поддерживающие Bluetooth, микроволновые печи и др., что ухудшает электромагнитную совместимость;
- производителями оборудования указывается скорость чуть большая, нежели реальная скорость передачи данных, т.к. реальная скорость зависит от доли служебного трафика, которая зависит уже от наличия между устройствами физических преград (мебель, стены), наличия помех от других беспроводных устройств или электронной аппаратуры, расположения устройств относительно друг друга и т.п.;
- частотный диапазон и эксплуатационные ограничения в различных странах не одинаковы;
- слабая стойкость алгоритма WEP-шифрования для устройств, поддерживающих лишь этот алгоритм шифрования.

7.2. Технология WiMAX (Worldwide Interoperability for Microwave Access)

Технология WiMAX — телекоммуникационная технология операторского класса, разработанная с целью предоставления беспроводной связи на большие расстояния с высоким качеством сервиса, основана на стандарте 802.16. Обеспечивает мультисервисность, гибкое распределение частот, задание приоритетов различным видам трафика, возможность обеспечения разного уровня трафика, поддержку интерфейсов IP, TDME1/T1. Технология WiMAX позволяет параллельно передавать голос, мультимедийную информацию и цифровые данные

по одному каналу (технология Triple Play). Важным преимуществом является возможность быстро наращивать емкость и расширить территорию связи. Базовые станции не требуют наличия высоких мачт, достаточно разместить их антенны на высоких зданиях или существующих мачтах высотой порядка 50 м. Согласно спецификации 802.16, максимальное расстояние, на котором возможно взаимодействие по сетям WiMAX, составляет 50 км, а суммарная пропускная способность — 70 Мбит/с. В условиях реальной эксплуатации эти показатели гораздо скромнее и составляют около 8 км и 2 Мбит/с [5].

Одна базовая станция в сети стандарта 802.16 может обслуживать большое количество пользователей и предоставлять им услуги разного уровня: например, для 60 бизнес-пользователей — услуги по каналу E1 (со скоростью 2,048 Мбит/с) и одновременно для сотен домашних пользователей с меньшими каналами требуемых частот.

Стандарт 802.16 обеспечивает высокий уровень конфиденциальности и безопасности сообщений, шифрование трафика в пределах всей беспроводной сети. WiMAX позволяет осуществлять доступ в Интернет на высоких скоростях и с гораздо большим покрытием, чем у Wi-Fi-сетей. Это позволяет использовать технологию для создания магистральных каналов, продолжением которых выступают традиционные DSL и выделенные линии. По этой же причине посредством WiMAX можно объединять и локальные сети удаленных офисов.

Структура сети WiMAX. Сеть WiMAX по своей архитектуре строится подобно сотовой сети, в основе которой лежит сеть базовых станций (BS). Каждая базовая станция по схеме «точка – много точек» может обслуживать с помощью всенаправленных антенн свою группу зданий в радиусе 6–8 км, образуя подобие ячейки сот.

При необходимости связи между удаленными ячейками базовые станции могут иметь направленные антенны и выполнять роль ретрансляторов по схеме «точка–точка» по радиоканалу на расстояниях до 50 км. С помощью ретрансляторов можно создавать региональные сети, состоящие как бы из островков локальных сетей. Доступ к глобальным сетям (например, общегородским, региональным и интернет-сетям) обеспечивается тем, что либо каждая базовая станция, либо одна из них, к которой через ретрансляторы или направленные антенны имеют доступ все остальные базовые станции, подключается проводным соединением или оптоволокном к магистральной сети. Такую базовую станцию называют точкой доступа к магистрали Backhaul. Схема такой архитектуры показана на рис. 7.2.

Антенны базовых станций могут быть установлены не только на мачтах, но и на крышах высоких зданий.

На первом этапе на обслуживаемых зданиях устанавливаются

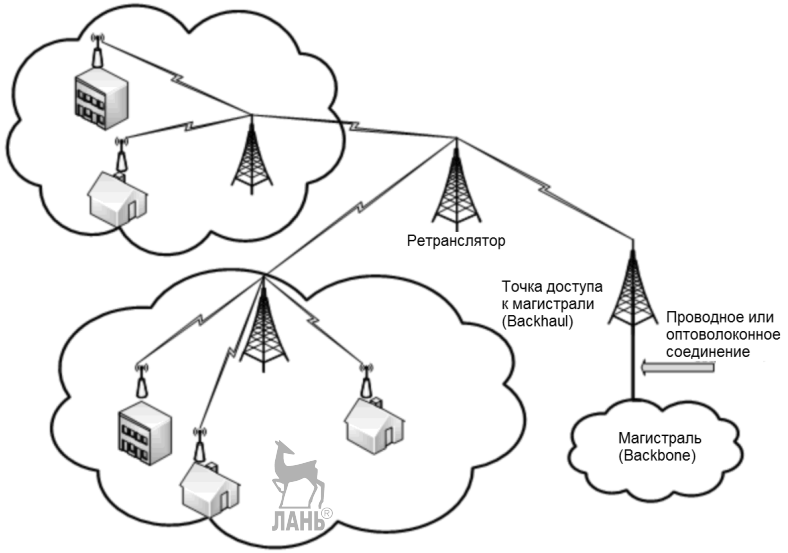


Рис. 7.2. Архитектура сети WiMAX

фиксированные наружные антенны, подключенные к блоку трансивера — станции клиентов находящемуся внутри здания. В блоке трансивера имеются стандартные проводные Ethernet-интерфейсы для подключения оборудования клиентов. Расположенные внутри здания ноутбуки, поддерживающие беспроводной стандарт 802.11, имеют в здании общую точку доступа (хотспот). Для организации выхода во внешнюю сеть трафик пользователей от различного оборудования объединяется с помощью мультиплексора, выход которого подключается к блоку трансивера клиентов и далее передается по сети WiMAX. Это позволяет сохранить существующую инфраструктуру районных или офисных локальных сетей при переходе с кабельного доступа на WiMAX, а также максимально упростить развертывание сетей [9].

Версии семейства WiMAX. Принято выделять фиксированный и мобильный варианты WiMAX. Каждая из спецификаций при этом определяет свои рабочие диапазоны частот, ширину полосы пропускания, мощность излучения, методы передачи и доступа, способы кодирования и модуляции сигнала и т.д. Фиксированный WiMAX позволяет обслуживать только «статичных» абонентов, а мобильный ориентирован на работу с пользователями, передвигающимися со скоростью до 150 км/ч. Мобильность означает наличие функции роуминга и «бесшовного» переключения между базовыми станциями при передвижении абонента.

802.16-2004 (известен также как 802.16d и фиксированный WiMAX). Спецификация утверждена в 2004 году. Используется ортогональное частотное мультиплексирование (OFDM), поддерживается фиксированный доступ в зонах с наличием либо отсутствием прямой видимости. Пользовательские устройства представляют собой стационарные модемы для установки вне и внутри помещений, а также PCMCIA-карты для ноутбуков. В большинстве стран под эту технологию отведены диапазоны 3,5 и 5 ГГц. Многие аналитики видят в ней конкурирующую или взаимодополняющую технологию проводного широкополосного доступа DCL.

802.16-2005 (известен также как 802.16e и мобильный WiMAX). Спецификация утверждена в 2005 году. Оптимизированная для поддержки мобильных пользователей версия поддерживает ряд специфических функций, таких как хэндовер и роуминг. Применяется масштабируемый OFDM-доступ (SOFDMA), возможна работа при наличии либо отсутствии прямой видимости. Частотные диапазоны для сетей Mobile WiMAX таковы: 2,3, 2,5, 3,4...3,8 ГГц.

Предоставленные варианты одной технологии неизбежно ставят потенциального клиента перед проблемой выбора, который, в свою очередь, зависит от множества факторов и должен осуществляться индивидуально в каждом конкретном случае. Тем не менее, можно проанализировать ключевые рыночные факторы и тенденции, которые способны повлиять на такой выбор [8].

Сначала отметим преимущества фиксированного WiMAX:

- более высокая пропускная способность за счет используемых типов модуляции и кодирования сигнала. Это очень важно для корпоративных пользователей, выдвигающих жесткие требования к полосе пропускания и имеющих возможность установки фиксированного модема;
- большой выбор поставщиков готового оборудования (спецификация 802.16d появилась раньше, чем 802.16e). Пользователи могут построить сеть, максимально соответствующую их специфическим требованиям;
- наличие оборудования, протестированного на взаимную совместимость. Сертификаты WiMAX Forum certified гарантируют полную совместимость устройств разных производителей.

В свою очередь, преимущества мобильного WiMAX таковы:

- поддержка мобильности за счет «передачи» установленного соединения между базовыми станциями (handover). Правда, при этом уменьшается радиус действия базовой станции, но такой недостаток нивелируется меньшей ценой BS. Предварительные расчеты показывают, что при создании сети масштаба

города на основе мобильного или фиксированного вариантов WiMAX стоимость базового оборудования оказывается примерно одинаковой;
- лучшее покрытие внутри помещений, которое обеспечивают метод SOFDMA и адаптивные антенны. Это очень важно, поскольку пользователи зачастую находятся внутри помещений, вне зоны видимости BS;
- создание субканалов по методу SOFDMA позволяет оператору более эффективно использовать выделенный частотный ресурс, предоставляя его только по запросу пользователей. Это весьма ценная возможность в условиях дефицита свободных частот;
- большой выбор абонентских устройств — модемов, карт для ноутбуков, смартфонов (спектр абонентских устройств для фиксированного WiMAX ограничен модемами и PCMCIA-картами). Это позволяет более точно дифференцировать услуги и адаптировать предложения беспроводных операторов к разным рыночным сегментам;
- возможно повторное использование частот в соседних сотах, что ведет к повышению эффективности применения выделенного частотного ресурса и избавляет от необходимости частотного планирования сети.

На выбор варианта WiMAX влияют бизнес-стратегия оператора, тип целевой аудитории, доступный частотный диапазон, ширина выделенных радиоканалов, особенности лицензирования и регулирования услуг связи. Архитектура создаваемой сети и возможности стандарта, положенного в ее основу, должны соответствовать стратегии бизнеса оператора. Если его целевой аудиторией являются корпоративные или домашние абоненты, которые могут установить внешнюю антенну и не нуждаются в мобильности, то правильный выбор — стандарт 802.16d. Если же значительной части клиентов требуется мобильность, то более подходящим окажется 802.16e. В большинстве случаев операторы широкополосного беспроводного доступа (ШБД) не могут использовать установленный диапазон по своему усмотрению и вынуждены довольствоваться полосами частот, определяемыми регулятором. А потому при выборе оптимального варианта технологии надо проанализировать доступность готовых продуктов для конкретного диапазона.

В представленной ниже таблице отображены основные свойства и классы двух рассмотренных версий стандарта WiMAX [7].

Wi-Fi и WiMAX. Несмотря на то, что технологии направлены на решение совершенно различных задач, очень часто их сопоставля-

Таблица 7.1. Общие свойства фиксированной и мобильной версий WiMAX

Свойство	Значение для клиента
Общемировой стандарт	Исключает зависимость от одного поставщика. Дает возможность экономить на масштабе сети
Высокая пропускная способность	Позволяет реализовать услуги, чувствительные к полосе пропускания
Большой радиус зоны действия	Позволяет экономично строить сети с обширным радиопокрытием
Единая IP-платформа	Позволяет внедрять широкий набор услуг и приложений на базе IP
Не требуется прямая видимость между точками	Подходит для условий городской застройки и сильно пересеченной местности
Обеспечение QoS в соответствии с классами обслуживания	Оптимизирует использование пропускной способности
Малая задержка при передаче в радиоканале	Позволяет внедрять приложения, чувствительные к задержкам
Высокая спектральная эффективность	Эффективное использование доступного радиочастотного ресурса

Таблица 7.2. Сравнение стандартов IEEE 802.11 и 802.16

Технология	Стандарт	Использование	Пропускная способность	Радиус действия	Частота
Wi-Fi	802.11a	WLAN как правило	до 54 Мбит/с	до 300 м	5,0 ГГц
Wi-Fi	802.11b	WLAN как правило	до 11 Мбит/с	до 300 м	2,4 ГГц
Wi-Fi	802.11g	WLAN как правило	до 54 Мбит/с	до 300 м	2,4 ГГц
Wi-Fi	802.11n	WLAN как правило	до 600 Мбит/с	до 300 м	2,4...2,5 или 5,0 ГГц
WiMAX	802.16d	WMAN	до 75 Мбит/с	25...80 км	1,5...11 ГГц
WiMAX2	802.16e	Mobil WMAN	до 40 Мбит/с	1...5 км	2,3...13,6 ГГц
WiMAX	802.16m	WMAN, Mobil WMAN	до 1 Гбит/с до 100 Мбит/с	120...150 км	

ют. Приведем сравнительную таблицу этих стандартов беспроводной связи.

WiMAX — это система дальнего действия, покрывающая километры пространства, которая обычно использует лицензируемые спектры частот. Разные стандарты 802.16 обеспечивают разные виды дос-

тупа, от мобильного (схож с передачей данных с мобильных телефонов) до фиксированного (альтернатива проводному доступу, при котором беспроводное оборудование пользователя привязано к местоположению). Wi-Fi — это система, охватывающая меньший радиус, обычно покрывающая десятки метров, которая использует нелицензированные диапазоны частот для обеспечения доступа к сети. Обычно это доступ к собственной мобильной сети пользователя, которая, может быть, и не подключена к Интернету. У технологий Wi-Fi и WiMAX совершенно разные механизмы Quality of Service (QoS). WiMAX использует механизм, основанный на установлении соединения между базовой станцией и устройством пользователя. При этом каждое соединение основано на специальном алгоритме планирования, гарантирующем параметры QoS для этого соединения. Wi-Fi же использует механизм QoS, подобный тому, что используется в Ethernet, при котором пакеты получают различный приоритет, что не гарантирует одинаковой QoS для каждого соединения [10].

7.3. Переход от WiMAX к технологии LTE (LongTermEvolution)

LTE Advanced вместе с WiMAX2 были официально признаны беспроводным стандартом связи четвертого поколения 4G Международным союзом электросвязи. LTE Advanced — стандарт мобильной связи 3GPP10 версии, направленный на улучшение стандарта LTE (Long Term Evolution). Технология LTE пережила целый ряд этапов развития с момента выхода первоначального стандарта, принятого консорциумом 3GPP, так называемого 3GPP Релиза 8. Для улучшения эксплуатационных характеристик и расширения возможностей технологии в апреле 2008 года консорциум 3GPP начал работу над Релизом 10. Одной из задач было достижение полного соответствия технологии LTE требованиям стандарта IMT-Advanced (официальный статус сетей 4-го поколения), установленного для 4G Международным союзом электросвязи, что позволило бы с полным правом назвать LTE технологией 4G [4].

LTE Advanced предусматривает расширение полосы частот, агрегацию спектра, имеет расширенные возможности многоантенной передачи данных, поддерживает функции ретрансляции сигнала LTE, а также развертывание гетерогенных сетей (HetNet).

В октябре 2012 года Yota первой в мире запустила технологию мобильной связи LTE Advanced на коммерческой сети. В запуске участвовало 11 базовых станций. В 2014 г. Мегафон запустил в пределах садового кольца Москвы сеть LTE Advanced с максимальной скоростью до 300 Мбит/с на загрузку к абоненту и 50 Мбит/с от абонента.

Несмотря на то, что и WiMAX и LTE Advanced относятся

Таблица 7.3. Характеристики LTE и LTE Advanced

Характеристики	LTE (релиз 8)	LTE Advanced	IMT Advanced
Пиковая скорость на линии вниз	300 Мбит/с	1 Гбит/с	1 Гбит/с*
Пиковая скорость на линии вверх	75 Мбит/с	500 Мбит/с	0
Максимальная спектральная эффективность вниз, бит/с/Гц	15	30	15
Максимальная спектральная эффективность вверх, бит/с/Гц	3,75	15	6,75

*В требованиях МСЭ указано: 1 Гбит/с для низкой мобильности и 100 Мбит/с для высокой мобильности.

к четвертому поколению связи (4G), принято считать, что основным преимуществом LTE является преемственность технологии 3G (UMTS/HSPA, HSPA+), а WiMAX же является отдельной ветвью эволюции мобильного широкополосного доступа. В будущем WiMAX имеет неблагоприятные рыночные перспективы, т.к. имея сопоставимые с LTE характеристики, технология оказалась менее приспособлена к массовому развертыванию.

Многие крупные операторы БШПД постепенно мигрируют на технологию LTE в непарном спектре (TD-LTE) и свертывают инвестиции в WiMAX. По различным прогнозам, на долю WiMAX в ближайшее время придется только около 1% рынка мобильного ШПД в целом и до 13% абонентской базы 4G в мире. Единственным оператором, который по-прежнему развивает WiMAX, является японский UQ Communications. Два других крупнейших WiMAX-оператора — американский Clearwire и азиатский Packet One — стратегически нацелены на переход к LTE. На сегодня в мире насчитывается несколько десятков коммерческих сетей TD-LTE. Существенным драйвером является выпуск устройств, поддерживающих оба стандарта — WiMAX и TD-LTE. Это позволяет операторам совершить плавную миграцию с одной технологии на другую без перерывов в обслуживании абонентов [4].

Сдерживают развитие WiMAX ограниченный ряд абонентских устройств, фактическое отсутствие роуминга и отказ крупнейших вендоров и мобильных операторов от инвестиций в эту технологию.

Описание сети радиодоступа RANLTE. Технология LTE относится к сетям сотовой связи. Это означает, что большая сеть разбивается на несколько *cot*, каждая из которых содержит базовую станцию. Такая базовая станция получила название *eNodeB*. Кроме того, внутри соты находятся оконечное оборудование/пользовательские устройства (UE, User Equipment) [2], такие как мобильные телефоны,

узлы беспроводной сенсорной сети, персональные компьютеры и т.д.

Архитектуру сети мобильной передачи данных LTE можно разделить на две основные части: RAN (Radio Access Network, сеть радиодоступа) и CN (Core Network, ядро сети), осуществляющие различные функции.

Сеть радиодоступа RAN ответственна за весь функционал, связанный с беспроводным доступом до ближайшей базовой станции — соединением и контролем за использование радиоресурсов мобильными терминалами (UE, User Equipment). Ядро сети (CN) осуществляет аутентификацию, биллинг и установление маршрута соединения [1].

Ядро сети, также известное как EPC (Evolved Packet Core), поддерживает доступ только с *коммутацией пакетов* (КП) и состоит из нескольких типов логических элементов (узлов) [1]:

- MME (Mobility Management Entity). Функция MME заключается в контроле мобильности пользователей — управление ключами безопасности, подключение и освобождение потока от UE, переключение режимов IDLE и ACTIVE мобильного терминала;
- S-GW (Serving Gateway) отвечает за предоставление соединения между eNodeB и UE. Кроме того, S-GW осуществляет обработку биллинговой информации и статистики, а также поддерживает другие технологии 3GPP (GSM/GPRS и HSPA);
- P-GW (Packet Data Network Gateway) предоставляет Интернет-доступ для мобильных терминалов. Данный шлюз ответственен за распределение IP-адресов, поддержание заданного уровня QoS, а также за мониторинг и подсчет интернет-трафика. Кроме того, P-GW поддерживает технологию CDMA 2000 для доступа в EPC;
- PCRF (Policy and Charging Rules Function) отвечает за управление уровнями QoS и тарифами биллинга;
- HSS (Home Subscriber Service) представляет собой базу данных, содержащую информацию о локальных пользователях;
- MBMS (Multimedia Broadcast Multicast Services). С помощью MBMS осуществляется поддержка различных сервисов multicast/broadcast в рамках единой сети. MBMS формирует специальную область для передачи потока данных множеству пользователей, подписанных на определенную услугу. Пример процесса передачи одного IP пакета через весь стек протоколов RAN изображен на рис. 7.3.

Сеть радиодоступа RANLTE представляет собой некоторое число базовых станций eNodeB, отвечающих за все беспроводные функции в зоне их действия. Базовая станция eNodeB соединяется с ядром

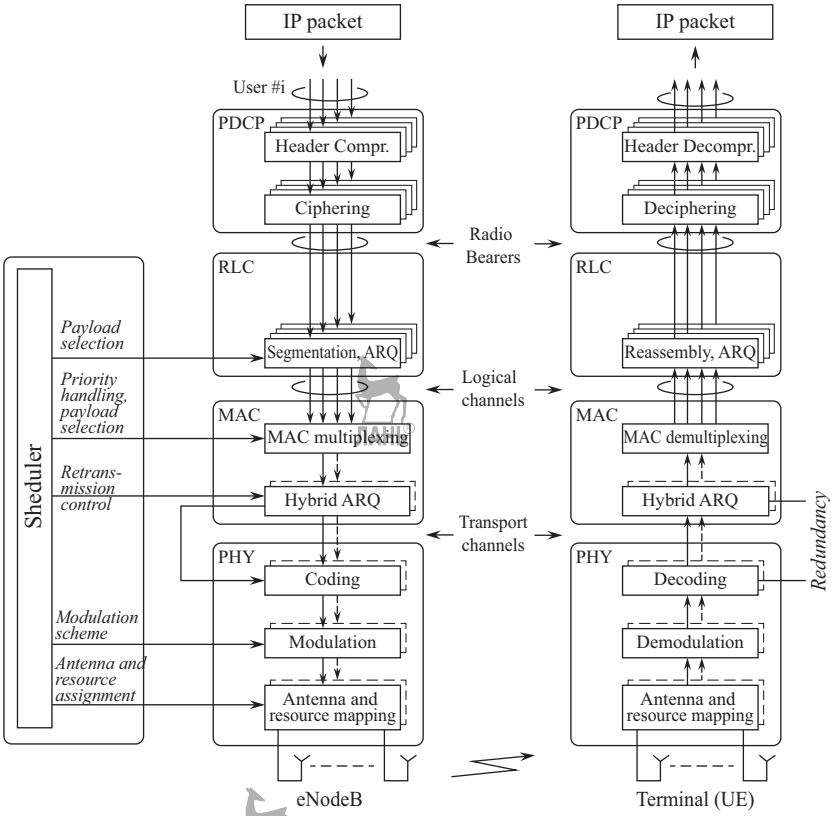


Рис. 7.3. Стек протоколов RANLTE

сети (EPC) посредством интерфейсов Sl-u и S1-c. Каждый eNodeB может быть соединен с несколькими узлами S-GW и MME в целях распределения нагрузки и улучшения структурной надежности.

Несколько eNodeB соединены между собой посредством X2-интерфейса, который используется для нужд управления радиоресурсами (например, для ICIC — Inter-Cell Interference Coordination), а также для пересылки пакетов внутри RAN.

Стек протоколов RANLTE. Обобщенный стек протоколов сети радиодоступа LTE состоит из нескольких протоколов/уровней (рис. 7.5). Это протоколы/уровни PDCP (Packet Data Convergence Protocol), RLC (Radio-Link Control), MAC (Medium-Access Control), PHY (Physical Layer).

Протокол PDCP осуществляет сжатие заголовка IP для уменьшения количества передаваемой информации. Кроме того, PDCP ответственно за шифрование/дешифрование и целостность передаваемых

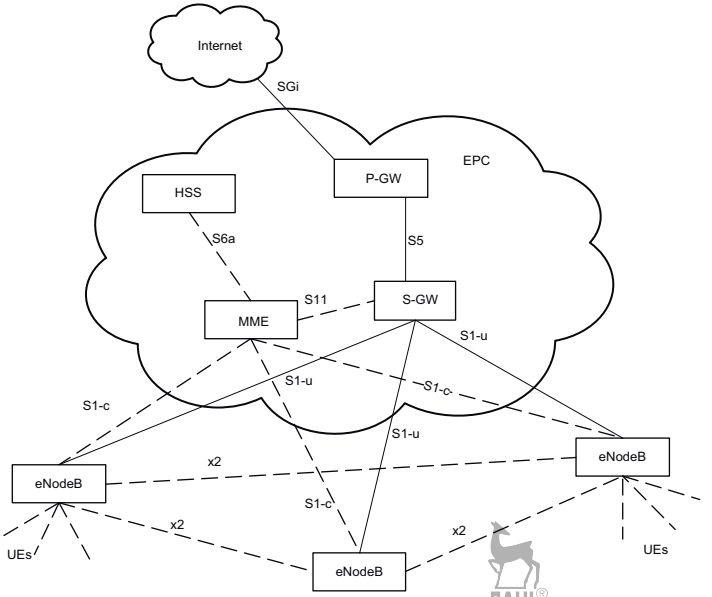


Рис. 7.4. Обобщенная архитектура сети LTE

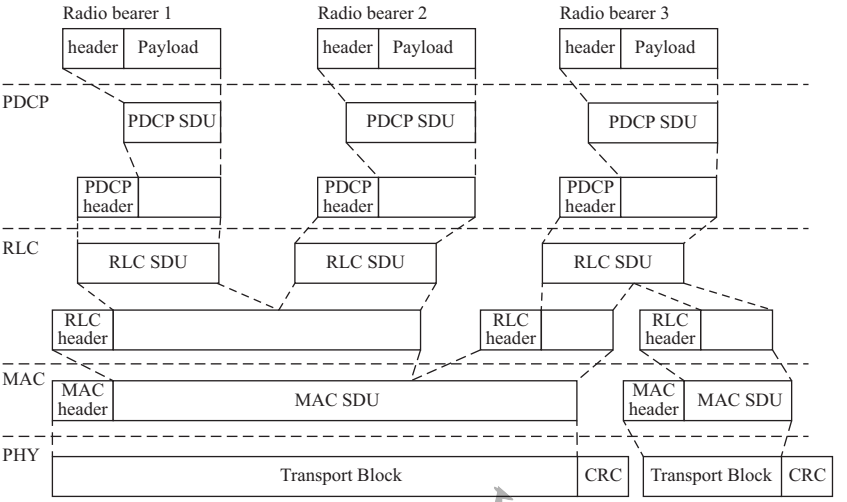


Рис. 7.5. Пример «обработки» IP пакета стеком протоколов RAN

данных, а также за правильный порядок получения пакетов и удаление дубликатов при хэндовере. Пакеты PDCP соответствуют одному радиопотоку для конкретного терминала.

RLC протокол несет ответственность за сегментацию/объединение, повторную передачу, обнаружение дубликатов и после-

довательную передачу данных на вышестоящие уровни. Каждый IP-пакет привязывается к конкретным радиопотокам, в зависимости от требований QoS.

Протокол MAC отвечает за формирование логических каналов при помощи мультиплексирования. Кроме того, MAC-уровень поддерживает ретрансмиссии Hybrid ARQ, а также распределяет uplink и downlink ресурсы.

Физический уровень осуществляет все необходимые действия: кодирование/декодирование, модуляция/демодуляция, многоантенное распределение (MIMO). На физическом уровне формируется так называемый транспортный канал, соединяющий UE и eNodeB.

RLC осуществляет сегментацию SDUPDCP и добавляет заголовок RLC, который используется для контроля последовательности в случае повторной передачи некоторого пакета. Далее RLCPDU пересылается на уровень MAC.

MAC-уровень осуществляет мультиплексирование нескольких RLCSDUs и добавляет MAC-заголовок, тем самым формируя *транспортный блок*. Количество RLCSDU в транспортном блоке зависит от механизма адаптации к условиям среды передачи.

На физическом уровне к RLCSDU добавляется блок CRC, затем осуществляется кодирование, модуляция и непосредственно передача сигнала.

7.4. Состояние и перспективы высокоскоростных беспроводных сетей

Увеличение пропускной способности беспроводных Wi-Fi сетей, с целью приближения их характеристик к проводным сетям (появление стандарта IEEE 802.11.ac), делает технологию более удобной и привлекательной для пользователя в существующих беспроводных сетях. Количество этих сетей в России увеличивается повсеместно. Сети Wi-Fi распространены сегодня везде — в школах, вузах, общежитиях, пешеходных зонах, на остановках транспорта и т.д. В 2013 году объем российского рынка публичных сетей Wi-Fi превысил 1,5 млрд. рублей, о чем говорится в исследовании консалтингового агентства J'son & Partners Consulting [12].

На территории нашей страны доминируют неуправляемые публичные Wi-Fi сети, где юридические лица предоставляют доступ в Интернет своим клиентам (сегмент B2B). Управляемые же сети (управление происходит из единого центра) по количеству хот-спотов занимают около 40%. Примером таких сетей может быть проект «Городской Wi-Fi», в рамках которого за несколько лет в Москве предполагается установить 8000 точек доступа (хот-спотов), что позволит обеспечить широкополосный доступ к сети практически во всех ме-

стах массового пребывания людей (в том числе на стадионах, где летом 2018 года будут проходить матчи Чемпионата мира по футболу) [11]. Разработкой и внедрением этого проекта занимается подведомственная Россвязи ФГУП РСВО («Российские сети вещания и оповещения»), а телеком-партнером проекта является китайская компания Huawei. Проект носит и социально значимый характер, так как расширяет возможности ФГУП РСВО по оповещению граждан о чрезвычайных ситуациях и общих угрозах. В рамках этого проекта к 1 января 2016 года хот-споты «Городского Wi-Fi» охватят все места массового пребывания людей, расположенные в Центральном административном округе Москвы, а к 2019 году они охватят всю столицу (в пределах ее старых границ).

Кроме того, операторы активно внедряют проекты по развертыванию Wi-Fi сетей на общественном транспорте. Сегодня такие сети функционируют на станциях и линиях метро в крупных городах России (Москва, Новосибирск и т.д.) и в поездах дальнего следования («Сапсан», «Аллегро» — ОАО «МегаФон»).

Несколько тысяч бесплатных точек доступа Wi-Fi обеспечивает и компания ОАО «МТС» более чем в 40 городах России от Калининграда до Южно-Сахалинска. По прогнозам того же агентства J'son & Partners Consulting, в 2015–2018 гг. рынок публичных сетей Wi-Fi будет расти показателем CAGR (Compound Annual Growth Rate — совокупный среднегодовой темп роста) равным 5%, и к 2018 г. его объем превысит 290 тысяч точек доступа [12].

Несмотря на то, что многие эксперты прогнозировали полный отказ в ближайшем будущем от WiMAX в пользу LTE, в России этого не произошло. Пока технология WiMAX развиваются параллельно с LTE и имеют хорошие перспективы в России.

Технология WiMAX используется, как правило, для подключения корпоративных клиентов, офисы которых не подключены к волоконно-оптическим линиям связи ввиду невозможности провести ВОЛС (Дальний восток и пр.) или высокой стоимости строительства, а также в районах малоэтажной застройки. Кроме того, для WiMAX есть ряд определенных ниш: системы безопасности, технологическая связь, интеллектуальные сети и т.д. [12].

Однако некоторые компании, предоставляющие услуги WiMAX, перешли на LTE-TDD (сети WiMAX демонтированы в Санкт-Петербурге, Москве компанией «Скартел», «Комстар»). Тем не менее география сетей WiMAX в России все же широка: Муром, Елец, Ухта, Энгельс, Канск, Ачинск, Рязань, Челябинск, Улан-Удэ, Благовещенск, Хабаровск — неполный перечень городов, где сети WiMAX (на конец 2014 — начало 2015 г.) обеспечивает беспроводной доступ в сеть Ин-

тернет на скорости до 300 Мбит/с.

Long Term Evolution (LTE) — это следующий значительный шаг в области широкополосного беспроводного интернет-соединения, занимающий все более крепкие позиции.

По оценкам ассоциации GSA (Globalmobile Suppliers Association), пик запуска новых сетей LTE в мире пришелся на 2012 г., в течение которого их количество увеличилось втрое. На начало июля 2014 г. в мире насчитывалось более трехсот коммерческих сетей стандарта LTE в 107 странах мира, среди них 13 — LTE Advanced [13]. И развитие это продолжается. 69 операторов в 36 странах мира инвестируют в сети LTE Advanced (конец 2014 г.), запускаются все новые и новые сети, повышая эффективность сетевой инфраструктуры. У всех ведущих операторов России (Ростелеком и «большая тройка») есть технологические возможности запуска сетей на основе технологии LTE Advanced (на основании действующих на территории России частотных назначений). И каждый из операторов стремится увеличить число вводимых в эксплуатацию сетей LTE. География таких сетей широка: Москва, Сочи, Новосибирск и т.д.

В уникальном положении при этом не только в России, но и во всем мире оказываются операторы «МегаФон» и «Скартел», объединившие свои ресурсы, располагая сплошной полосой спектра LTE шириной 40 МГц. Это позволило запустить самую быструю в мире сеть на основе технологии LTE-A, которая действует в Москве в пределах Садового кольца, на олимпийских объектах в Сочи под торговой маркой 4G+. Доступные скорости в этих сетях составляют до 300 Мбит/с.

Одним из факторов, влияющих на развитие современного рынка беспроводных технологий, является повышение эффективности использования радиочастотного спектра и увеличение скорости передачи данных. Появляющиеся новые стандарты рассмотренных выше технологий как нельзя лучше этому соответствует.

7.5. Контрольные вопросы

1. Назовите преимущества Wi-Fi.
2. Как организована сеть Wi-Fi?
3. Расшифруйте аббревиатуру WiMAX. Для каких целей разрабатывалась эта технология?
4. Назовите основные версии семейства WiMAX и их характеристики.
5. Сравните технологии WiMAX и Wi-Fi.
6. Назовите преимущества LTE Advanced перед LTE.
7. Перечислите функции ядра сети (CN).
8. Опишите сеть радиодоступа RANLTE.

7.6. Список литературы

1. *Dahlman E., Parkvall S., Skold J.* 4G: LTE/LTE-Advanced for Mobile Broadband. — Academic Press, 2011. ISBN 978-0-12-385489-6.
2. *Michael Burakov, Albin Eldstål-Damlin* An LTE Random Access Channel Model for Wireless Sensor Network Applications: Master Thesis. — Lulea University of Technology, Sweden, 2012, 66 P.
3. *Бакланов И.Т.* NGN: принципы построения и организации / под.ред. Ю.Н. Чернышова. — М.: ЭкоТрендз, 2008. — 400 с.
4. *Вишневский В.М., Портной С.Л., Шахнович И.В.* Энциклопедия WiMAX. Путь к 4G: монография. — М.: Техносфера, 2009. — 470 с.
5. *Гордейчик С.В., Дубровин В.В.* Безопасность беспроводных сетей. — М.: Горячая линия–Телеком, 2008. — 288 с.
6. *Беделл П.* Сети. Беспроводные технологии. — НТ Пресс, 2008. — 448 с.
7. *Широков В.* Методология беспроводных мультисервисных сетей класса WiMAX. Ч.1 // Технологии и средства связи. — 2010. — № 1. — С.32–33; 2010. — № 5. — С.36–37.
8. *Николаев В., Гармонов А., Лебедев Ю.* Системы широкополосного доступа 4 поколения // Первая миля. — 2010. — № 5/6. — С.56–59.
9. *Шахнович И.* Архитектура сети WiMAX: основные элементы и принципы // Первая миля. — 2009. — № 1. — С.6–15.
10. *Иванов М.* Беспроводные технологии: Wi-Fi и WiMAX [электронный ресурс] // poisk-podbor.ru: информ.-справочный портал. М., 2008–2015. http://routers.poisk-podbor.ru/article/articles/besprovodnye-tehnologii-wi-fi-i-wimax/19.html (дата обращения 28.01.15).
11. Московский «Городской Wi-Fi» построят на радиотрансляционной сети // Сети связи: строительство управление, модернизация. — 2014. — № 10(58). — С.36–37.
12. Telekomza. Кучумова А. WiMAX в России еще послужит [электронный ресурс] // telekomza.ru: статья от 1.02.13 URL: http://telekomza.ru/2013/02/01/wimax-v-rossii-eshhe-posluzhit.html (дата обращения: 3.03.2015).
13. Fr.slideshare [электронный ресурс] // http://fr.slideshare.net/: информационный портал. URL: http://fr.slideshare.net/RadikalTR/gsa-evolution-to-lte-report-050112 (дата обращения: 3.03.2015).

Глава 8

Вместо заключения: некоторые соображения на тему «что надо сделать, чтобы обеспечить передачу данных с высокой скоростью в IP-сетях»

Задача передачи данных между сетевыми узлами возникла почти одновременно с появлением первых ЭВМ. Но представление о быстрой передаче данных по IP-сетям в разное время было разным. Это связано преимущественно с тем, что и отношение к объёмам данных претерпевало изменения. На заре компьютерной эры данные измерялись байтами и килобайтами, а мегабайтами измерялся совокупный объем цифровых данных на Земле. Понятно, что задачей первых IP-сетей было обеспечение передачи данных на скоростях, соизмеримых с объёмом передаваемых данных — биты и килобиты в секунду.

Стремительное развитие цифровой техники увеличивает объёмы данных, с которыми сталкиваются пользователи, что обуславливает необходимость к увеличению скорости передачи этих данных. В наше время скорость в несколько десятков мегабит доступна многим даже на домашних терминалах, например, сжатому видеопотоку в формате FullHD необходимо порядка 10 Мбит/с полосы пропускания. Активно внедряется услуга доставки видео контента в формате 4k, для чего требуется уже не менее 15 Мбит/с. Такие скорости были немыслимы в домашних условиях всего лишь одно десятилетие назад. В профессиональной среде необходимость в доступной пропускной способности достигает сотен гигабит в секунду.

Из сказанного выше ясно, что понятие «высокоскоростная передача данных» имеет неразрывную связь со временем. Так, высокоскоростная передача данных 30 лет назад — это килобиты в секунду; сейчас для большинства случаев это десятки и сотни мегабит. Не исключено, что через 10 лет контент будет доставляться до терминала пользователя со скоростями, превышающими 100 Мбит/с.

В свете этого целесообразно ввести такое понятие, как высокоэффективная передача данных. Под высокоэффективной передачей данных подразумевается, что скорость передачи информации близка к пропускной способности соединения, несмотря на наличие задержек и потерь пакетов.

8.1. Традиционная передача данных с гарантированной доставкой. Проблемы

Традиционным протоколом передачи данных с гарантированной доставкой контента в IP-сетях является протокол TCP. Этот протокол был представлен впервые около 40 лет назад — в декабре 1974 года. Он был разработан для передачи данных в инфраструктуре тех дней, когда сети ограничивались лишь локальными сетями. Задача передавать данные по IP-сетям на большие расстояния возникла позже. А передача данных со скоростями в несколько гигабит в секунду на большие расстояния стала актуальной лишь в последнее десятилетие. Безусловно, алгоритмы протокола TCP со временем претерпевали изменения. Было представлено множество версий протокола, предназначенных для использования в различных условиях. Самая стабильная и наиболее используемая в наше время версия TCP, оптимизированная для соединений с большой пропускной способностью, называется «cubic».

Эксперимент, поставленный в техническом университете Мюнхена (TUM) демонстрирует изменение скорости передачи данных посредством протокола FTP по TCP (cubic) между двух сетевых узлов в зависимости от параметров соединения. Результат эксперимента представлен на рис. 8.1.

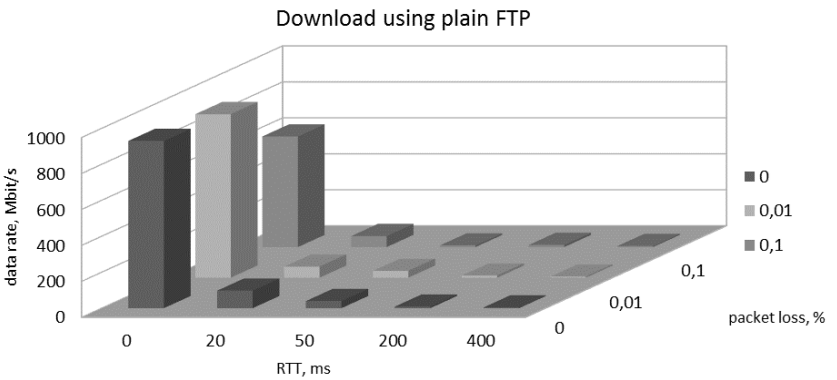


Рис. 8.1. Скорость передачи данных FTP/TCP

В ходе эксперимента использовался канал с пропускной способностью 1 Гбит/с. Эмулировались дополнительные задержки до 400 мс RTT (Round Trip Time) и потери пакетов до 0,1%. В расчётах обычно исходят из того, что один километр оптического волокна задерживает сигнал на 5 мкс. Таким образом, если пренебречь задержкой в промежуточных узлах, 10 мс соответствуют примерно 2000 км оптических линий. Соответственно, передача данных по каналу с RTT 20 мс

отображает соединение с узлом, который удалён ориентировочно на 4000 км. Как легко заметить из рис. 8.1, уже при такой задержке без потерь пакетов скорость передачи данных падает приблизительно на 90%, по сравнению с передачей данных без дополнительной задержки и потерь между узлами. С увеличением задержки между узлами скорость передачи падает ещё сильнее: уже при RTT = 50 мс используется только 5% пропускной способности канала.

На скорость передачи данных пагубно влияют также и потери пакетов. Примечательно, что даже в отсутствие дополнительной задержки при потере пакетов 0,1% скорость передачи данных снижается до отметки в 600 Мбит/с. При потере пакетов и дополнительной задержке скорость передачи данных уменьшается ещё больше; так, при 20 мс RTT и 0,01% потерь пакетов скорость передачи составляет порядка 60 из 1000 Мбит/с.

Величина потерь пакетов в IP-сетях — это сложный феномен, который невозможно описать с помощью некоего универсального подхода. В своей статье [1] Паксон показывает неоднородность сетевого параметра потери пакетов. В соответствии с этой статьёй, потери пакетов между 35 узлами в сети Интернет в 9 странах в 1994 году были не менее 2,7%. Эти данные имеют мало общего с современными сетями, но дают представление о том, что соединения типа WAN (Wide Area Network) характеризуются высоким уровнем потерь пакетов. Более актуальную информацию можно вынести из исследования [2] от 2009 г., где Ванг с соавторами приводят материалы исследований порядка 10 000 различных соединений в 147 странах. Авторы показывают, что около 10% всех исследуемых внутриконтинентальных соединений в Европе и Северной Америке имели уровень потери пакетов более 2%. В Азии, Африке, Южной Америке и Океании этот показатель составил не менее 20% (рис. 8.2).

Основываясь на вышеизложенном, можно утверждать, что показатель потери пакетов на уровне 0,1% может встретиться в реальном соединении.

8.2. Альтернативные протоколы передачи данных с гарантированной доставкой

Итак, TCP показывает себя хорошо на соединениях типа LAN (малый RTT и низкий показатель потерь пакетов), и имея в виду тот факт, что этот протокол используется почти всеми сетевыми узлами, в локальных сетях удобно использовать именно его. Неспособность TCP обеспечивать высокоэффективную передачу данных на широкополосных сетях с большой задержкой породила свободную нишу на рынке. В виду возросшего спроса на передачу больших объёмов данных, в последнее десятилетие эта ниша была заполнена как ком-

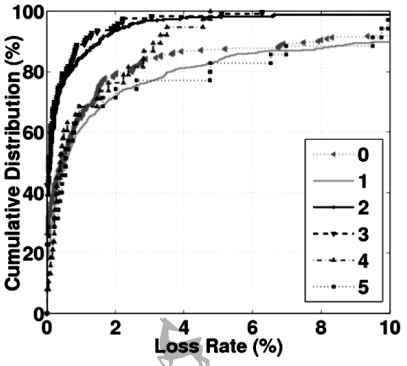


Рис. 8.2. Статистика потерь пакетов. 0 — Африка, 1 — Азия, 2 — Европа, 3 — С. Америка, 4 — Океания, 5 — Ю. Америка [2]

мерческими продуктами, так и приложениями с открытым исходным кодом.

В лаборатории FILA (Future Internet Lab Anhalt) были поставлены эксперименты с 9 современными протоколами и приложениями для высокоэффективной передачи данных.

Для проведения экспериментов в лабораторных условиях была создана сеть с использованием эмулятора канала Apposite Netropy 10G. Данный сетевой эмулятор позволяет на скоростях до 21 Гбит/с вводить в сеть до 10 секунд задержки, при необходимости обеспечивая её вариацию (джиттер), а также любое относительное значение потерь пакетов. Характерной особенностью эмулятора является его способность воспроизводить RTT с точностью до десятков наносекунд, что выгодно выделяет его на фоне сетевых эмуляторов, базирующихся на пакете ядра Linux — netem, где точность измеряется в десятках микросекунд. Использование сетевых эмуляторов в лабораторных условиях оправдано их относительно низкой стоимостью по сравнению с реальными каналами, а также высокой точностью эмуляции помех в канале. В публикации [3] Сеттелмаер с соавторами используют сетевой эмулятор для воспроизведения характеристик реальных 10 Гбит/с каналов, которые были у них в наличии. Сравнительный анализ эмулированной и реальной сетей показал, что расхождения в поведении незначительны. В заключение авторы обосновывают эффективность использования сетевых эмуляторов и подчёркивают удобство их использования.

В статьях [4] и [5] приведены результаты эксперимента с использованием альтернативных решений для обеспечения высокоэффективной передачи данных по широкополосным каналам. При этом

использованы коммерческие продукты:

- TIXStream, продукт фирмы TIXEL GmbH, ФРГ; основан на протоколе RWTP;
- ExpeDat от Data Expedition Inc, США; основан на протоколе MTP;
- Velocity, от производителя BitSpeed LCC, США;
- FC Direct — Unlimi-Tech Software Inc, Канада;
- Catapult — XDT Pty, Ltd, Австралия.

Протоколы для высокоэффективной передачи данных:

- RBUDP, протокол с открытым программным кодом;
- UDTv4, протокол с открытым программным кодом;
- MTP, закрытый протокол компании Data Expedition Inc;
- RWTP закрытый протокол компании TIXEL GmbH.

Проведение эксперимента в два этапа (эксперимент с продуктами и эксперимент с протоколами) обусловлено тем, что приложение отличается от протокола передачи данных, как минимум, ещё чтением и записью данных. А это уже зависит не от коммуникации между узлами, а от реализации вызовов на чтение и запись в приложениях.

Топология лабораторной сети, на которой проходили эксперименты, изображена на рис. 8.3. В ней использованы два сервера, обладающих высокой производительностью, чтобы избежать локальных узких мест на машинах, 10-гигабитный Ethernet-свитч Extreme Network x650, сетевой эмулятор. Все сетевые узлы соединены оптическими волокнами.

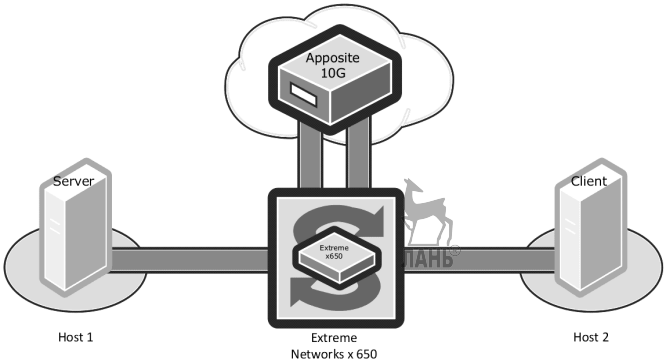


Рис. 8.3. Топология экспериментальной сети

На рис. 8.4 проиллюстрировано изменение скорости передачи данных в зависимости от задержки для указанных выше решений и протоколов для скоростной передачи данных. Графики демонстрируют, что в сетях с большими задержками все альтернативные решения обеспечивают лучшую скорость передачи, чем TCP.

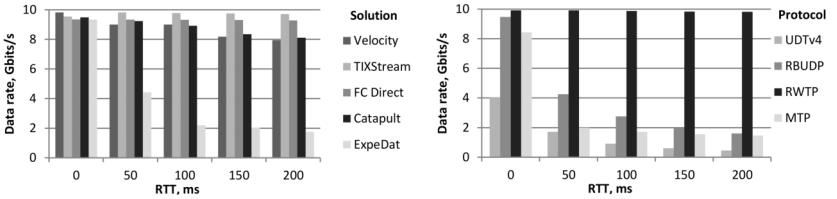


Рис. 8.4. Скорость передачи данных в зависимости от задержки в сети; без потерь пакетов

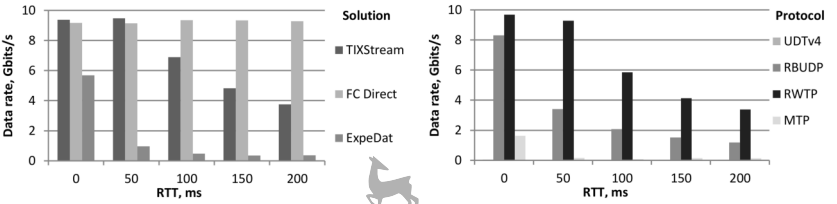


Рис. 8.5. Скорость передачи данных в зависимости от задержки в сети; потери пакетов: 1%

На сети с введённым высоким уровнем потерь пакетов — 1% (рис. 8.5) высокую производительность показывают не все решения. Так, на графиках не представлены решения Catapult, Velocity и протокол UDTv4, потому что средняя скорость передачи данных за сессию была ниже 100 Мбит/с.

Из графиков можно заключить, что уже сейчас существуют алгоритмы, которые способны передавать данные из конца в конец эффективно, невзирая на задержки и потери пакетов в сети. Но что мешает повсеместно использовать эти алгоритмы в клиентских терминалах?

Во-первых, исторически сложилось, что доступ к сокетам в разных операционных системах разный. Так, очень высокие показатели достигаются на UNIX-подобных операционных системах Linux. А вот наиболее популярная среди пользователей Microsoft Windows взаимодействует с сокетами на порядок медленнее. Примечательно, что некоторые из производителей альтернативных решений выпускают версии только для Linux ОС.

Вторая проблема, относящаяся к высокоэффективным решениям, — это их способность сосуществовать с другими участниками сети (fairness). Особенности многих альтернативных протоколов, связанные с тем, что они работают поверх UDP, делают их довольно агрессивными. Они занимают почти весь канал, вытесняя почти весь другой трафик. Таким образом, эти решения будут эффективно выполнять свою задачу — передавать данные, но используемые каналы будут недоступны для других приложений на время, необходимое для

высокоскоростной передачи данных.

Ключевым моментом для альтернативных высокоэффективных решений для высокоскоростной передачи данных является тот факт, что они должны работать в существующей инфраструктуре. Другими словами, чтобы все сетевые устройства пропустили трафик, необходимо использовать существующие протоколы. В противном случае затраты на обновление инфраструктуры будут неоправданно велики. Исследованные решения работают поверх UDP или TCP, а некоторые используют оба протокола: UDP для передачи данных, а TCP — для контроля соединения.

Безусловно, в случае передачи данных по UDP вся логика гарантированной доставки данных должна быть реализована отдельно, потому что UDP не предоставляет такую возможность.

Протокол UDP используется следующими приложениями: TIXStream (RWTP), FileCatalyst Direct, ExpeDat (MTP), RBUDP, UDTv4.

Catapult и Velocity используют для передачи данных TCP. Но TCP уже имеет свою систему контроля за перегрузками и логику повторной передачи потерянных пакетов. Соответственно, как только в сети появляются потери, эти решения не могут больше обеспечивать высокую эффективность.

Решением передачи большого объёма данных, используя TCP, зачастую является использование многопоточности. В представленном эксперименте Velocity и Catapult — решения, основанные на многопоточном TCP. В этом случае для передачи данных создаётся не одно, а сразу несколько соединений из конца в конец. Такой подход позволяет без изменений в инфраструктуре создать много потоков, каждый из которых передаёт данные с относительно низкой скоростью, но суммарная мощность потоков будет существенной. Другими словами, этот подход можно описать как использование множества низкоскоростных потоков вместо одного высокоэффективного. Достоинством такого подхода является дешевизна решения. Недостатки — это использование дополнительных системных ресурсов на конечных устройствах, искусственная конкуренция между потоками одной транзакции в сети, сложности в настройке файервола (такому соединению необходимо обеспечить сразу серию портов из конца в конец). В итоге эффективность передачи данных на сетях с потерями все равно ниже, чем у альтернативных решений, основанных на UDP, однако выше, чем у однопоточного TCP потока.

8.3. Алгоритм контроля перегрузок

Контроль перегрузок из конца в конец для трафика, доставляемого с наибольшими усилиями (best-effort traffic) необходим для того,

чтобы избежать коллапса в глобальной сети.

Причины низкой эффективности TCP хорошо описаны Гу с соавторами в [6]. В основном они касаются именно алгоритма контроля перегрузок, основанного на идее скользящего окна (window-based), который используется в TCP. Алгоритм AMID (аддитивное увеличение, мультипликативное уменьшение) требует много времени для определения доступной пропускной способности; более того, именно он ответственен за низкую эффективность TCP при наличии потерь пакетов в канале: любые потери расцениваются как перегрузка и скорость передачи существенно падает. Этот алгоритм сокращает окно вдвое, если в ходе передачи была обнаружена потеря пакетов, в противном случае он увеличивает окно на 1 пакет.

В литературе показано, что большая задержка на сетях с большой пропускной способностью также снижает эффективность алгоритма контроля перегрузок, основанного на скользящем окне [7]. В статье показано, что в конечном итоге, если канал поделён между несколькими TCP-соединениями, то TCP-поток, в котором RTT из конца в конец меньше, будет иметь преимущество, т.е. будет использовать больше полосы пропускания, чем тот, где RTT больше.

Альтернативой алгоритму, основанному на скользящем окне, выступает алгоритм, основанный на скорости передачи (rate-based). В этом случае пакеты посылаются не непосредственно друг за другом при наличии свободных мест в окне, а с заранее определённым межпакетным временным интервалом. Таким образом, достигается передача данных с постоянной скоростью (constant bit rate). По обратной связи к передатчику от приёмника передаётся информация, с помощью которой корректируется скорость передачи. В этом случае решение о снижение скорости выносится не только на основании произошедших потерь пакетов, но и на основании изменения скорости приёма данных на принимающем узле. Такой подход может обеспечивать решение проблем, возникающих от перегрузок, с единичными потерями в сети, когда пакеты отбрасываются сетевыми устройствами из-за переполнения буферов. Более того, при грамотном использовании обратной связи возникает возможность избежать перегрузки как таковой. Можно обнаружить, что буферы в промежуточных сетевых устройствах заполняются, и заблаговременно снизить скорость передачи данных.

Оправдывает себя также схема обратной связи, в соответствии с которой подтверждения (acknowledgment) посылаются не на каждый пакет, а сразу для группы пакетов. В этом случае сообщения об успешно принятых пакетах посылаются периодически, например, с интервалом в 0,01 секунды. В случае, если приёмник обнаружил

потерю пакета или иную причину, ввиду которой пакет должен быть послан ещё раз, он посылает NAK (negative acknowledgment) — уведомление о необходимости повторной передачи. Это позволяет быстрее реагировать на проблемы в передаче данных, не дожидаясь окончания времени таймера на передачу.

8.4. Условия обеспечения передачи данных с высокой скоростью

Для обеспечения высокоскоростной передачи данных необходимо правильно выбрать сетевую инфраструктуру:

- физическую среду;
- сетевое оборудование:
- протоколы передачи данных.

Выбор физической среды. Выбор физической среды чрезвычайно сильно зависит от конкретного случая, то есть от того, что представляет собой проект сети и в каких условиях сеть должна функционировать. Рис. 8.6 представляет типы сред, чаще всего используемых в высокоскоростных сетях.

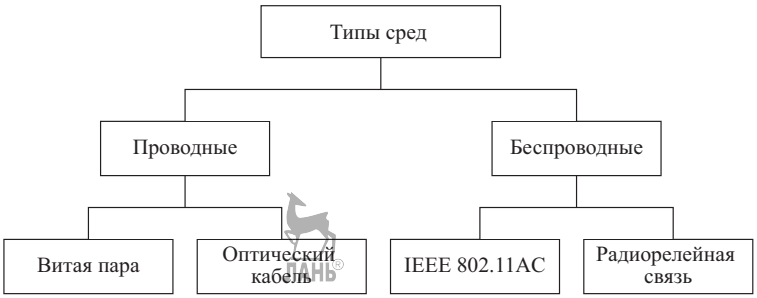


Рис. 8.6. Виды сред, используемых для высокоскоростной передачи данных

Для того, чтобы лучше понять обоснование выбора того или иного типа среды, предлагается рассмотреть три наиболее часто встречающихся сценария высокоскоростной передачи данных:

Ядро сети вычислительного центра (Data Center Network Core). Характеризуется следующими особенностями:

- чрезвычайно высокая связность сети;
- низкие расстояния между узлами сети;
- гибкость топологии — возможность перестроить или адаптировать сеть в кратчайшие сроки.

Магистральная сеть (провайдер). Примером может служить соединение между городами. Для данного случая характерны следующие особенности:

- малая связность;

– статическая топология (магистральная сеть не меняется с течением времени за исключением полосы пропускания);
– большие расстояния между узлами сети (например, населёнными пунктами).

Корпоративная сеть. Внутренние бизнес-процессы крупных компаний часто подразумевают собой передачу массивов данных внутри корпоративной сети. Такая сеть характеризуется следующими особенностями:

– высокая мобильность клиентских устройств (в пределах ограниченной территории);
– большое количество клиентских устройств.

Особые случаи. Примером может служить создание радиорелейного канала связи в мегаполисе или в труднодоступном регионе. Такая сеть, как правило, создается при невозможности использования стандартных методов.

Далее будут рассмотрены типы физических сред, применяемых в каждом отдельно взятом случае.

Ядро сети вычислительного центра. Вычислительный центр (ВЦ), как правило, отвечает высочайшим стандартам надёжности и качества связи. Каждый крупный вычислительный центр проходит сертификацию на соответствие определённому TIER-уровню. После сертификации ВЦ получает уровень от 1 до 4. Но даже в худшем случае TIER 1 время простоя в год не может превышать 1729,224 минуты, а в случае TIER 4 — не более 26,28 минут. Из года в год сотни людей работают над улучшением алгоритмов резервирования соединений, но и по сей день любой из подобных алгоритмов основан на высокой связности сети. Таким образом, в случае с ВЦ мы имеем высочайшую связность узлов на минимальной территории (чаще всего, в пределах одного здания).

Узлы в ВЦ чаще всего соединяются соединениями 10 Гбит/с с использованием витой пары категории 6 (10GBASE-T Cat.6), которая обеспечивает подобные скорости. Использование данного типа кабеля в ВЦ позволяет прокладывать линии связи длиной до 37 м (даже при условии группировки множества кабелей в один кабель-канал). Теоретический же предел (в соответствии со стандартом) составляет 100 м при условии отсутствия стороннего излучения от других кабелей. Разумеется, длина в 37 метров не является достаточной для ВЦ, поэтому необходимо использовать оптические кабели, которые, в свою очередь не накладывают таких строгих ограничений на длину сегмента сети.

Возникает вопрос, почему не использовать лишь оптические ка-

бели при построении сети? Ответ кроется в двух основных причинах:

- стоимость оптического кабеля и коннектора значительно выше по сравнению с витой парой;
- сложность в работе, обусловленная более высокой хрупкостью оптического кабеля и коннекторов.

Итак, ВЦ характеризуются использованием гибридной модели физической среды передачи данных, состоящей из оптического кабеля и витой пары категории 6.

Магистральная сеть. Топология магистральной сети несколько проще, чем в случае с ВЦ, однако обладает основной особенностью — чрезвычайно длинными сегментами сети (в наиболее общем случае, расстояние между населёнными пунктами). Таким образом, в современных магистральных сетях отпадает использование чего-либо, кроме оптического кабеля, который может с лёгкостью передать сигнал на десятки (и даже сотни) километров без использования какого-либо промежуточного оборудования.

В настоящее время (2015 год) ведётся множество исследований, посвящённых передаче данных посредством использования лазера. Наибольших успехов добились Япония и Дания, однако ни одна из разработок пока не вышла за пределы рабочего прототипа.

Итак, стандартом «де-факто» для магистральных сетей на сегодняшний день является связь по оптической линии.

Корпоративная сеть. Повсеместное распространение высокопроизводительных беспроводных устройств, а также развитие всем известного семейства стандартов IEEE 802.11 (Wi-Fi), привело к тому, что наличие беспроводной сети является необходимым фактором, обеспечивающим работу современного предприятия. Долгое время наличие лишь проводного доступа в компьютерную сеть воспринималось совершенно нормально, однако в данный момент многие внутренние процессы компаний выстроены вокруг возможности персонала развернуть своё рабочее место в любом месте рабочей сети, кампуса. В 2014 году завершилась работа над стандартом IEEE 802.11AC, который «перевернул» понимание стандартов беспроводной связи. Имея теоретический предел в 8 Гбит/с, данный протокол может полностью заменить классические проводные LAN-сети на полностью беспроводные. Из преимуществ подобного решения стоит отметить множество компаний, занимающихся реализацией «умных» Wi-Fi сетей, способных самостоятельно подстраиваться под изменяющийся трафик, масштабироваться и резервироваться. Однако в связи с недавним выходом данного стандарта необходимо время, чтобы плотно закрепиться на рынке, поэтому беспроводные сети сейчас работают совместно с проводными на корпоративном рынке, являясь лишь дополнением в

классической проводной LAN-сети предприятия.

На рынке корпоративных сетей используется гибридная модель на основе «витой пары» и беспроводного стандарта связи IEEE 802.11 (Wi-Fi) с преимущественным использованием проводных соединений.

Особые случаи. Возникают ситуации, когда прокладка проводного соединения невозможна по ряду причин, таких как:

- слишком сжатые сроки (форс-мажор);
- невозможность прокладки кабеля, например, в условиях гористой местности.

В таких случаях возможно применение радиорелейных линий (РРЛ), которые могут обеспечивать передачу данных на чрезвычайно высоких скоростях. Однако в силу своей специфичности, данный вид связи зачастую рассматривается как «запасной вариант». Так, к примеру, широкое распространение РРЛ получили в нефтяной индустрии, где данный вид связи используется на резервных каналах — в дополнение к оптической линии.

Вышеизложенное иллюстрирует рис. 8.7.

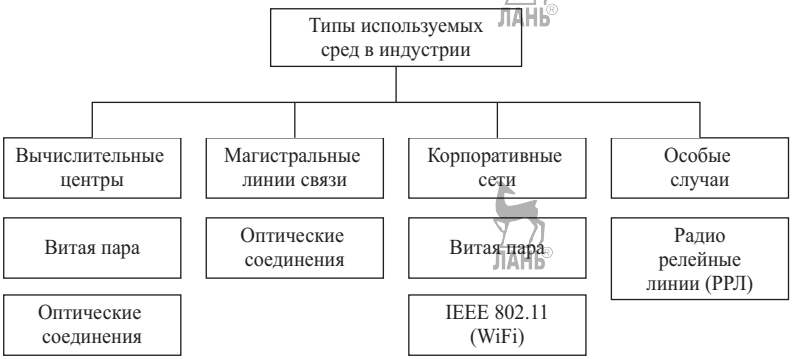


Рис. 8.7. Виды сред применяемых для высокоскоростной передачи данных в зависимости от поставленной задачи

Выбор сетевого оборудования. Непосредственно после принятия решения о типе физической среды в высокоскоростной сети, наступает очередь определиться с коммуникационным оборудованием: в общем случае это коммутаторы (switch) и маршрутизаторы (router). На рынке представлено множество производителей, и в соответствии с законами экономики, каждый из них представляет себя в лучшем свете. Вопрос выбора производителя (vendor) зачастую ложится на плечи проектировщика сети и представляет собой нетривиальную задачу, особенно в случае с постоянно эволюционирующей областью высокоскоростной передачи данных. При выборе производителя сетевого оборудования следует руководствоваться нижеперечисленными

факторами:

1. Личный опыт проектировщика и команды поддержки сети. Большинство современных производителей выпускают собственные операционные системы, устанавливаемые на коммутационное оборудование. Исторически сложилось так, что единого интерфейса не существует, и поэтому опыт работы команды с определенным оборудованием весьма важен как при ускорении процесса запуска сети, так и для оптимизации процессов поддержки.

2. Наличие полноценной технической поддержки от производителя: высокоскоростная передача данных чаще всего требуется там, где своевременная доставка информации является едва ли не определяющим фактором. Некоторые производители предлагают не только поддержку, но и экстренную замену вышедшего из строя оборудования, что является важным при необходимости минимизации времени простоя сети (downtime).

3. Поддержка последних технологий масштабирования и резервирования. Данный пункт особенно важен при проектировании ядра сети, где требуется высокая надёжность и связность узлов передачи данных.

4. Совместимость нового оборудования с уже существующим: проблемы совместимости чаще всего не затронут телекоммуникационную сеть в классическом понимании этого термина. Однако следование концепциям SDN (Software Defined Networks) накладывает чрезвычайно высокие требования на тип используемого оборудования в контексте программного обеспечения.

Надо отметить, что при проектировании высокоскоростной сети следует учесть несколько ключевых моментов, без учета которых высокоскоростная передача данных не представляется возможной. В целом, основное влияние на обеспечение высокоскоростной передачи данных оказывают 4 фактора, отображенные на рис. 8.8.

Среда передачи была подробно рассмотрена ранее, поэтому перейдем к остальным трем факторам.

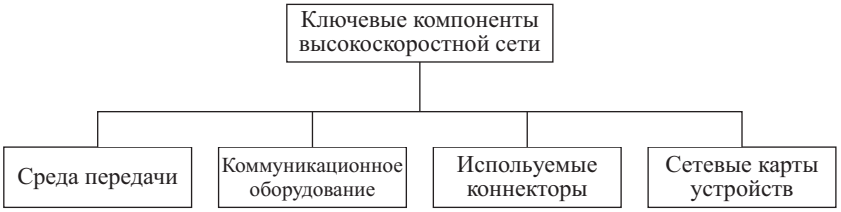


Рис. 8.8. Основные факторы, оказывающие влияние на обеспечение высокоскоростной передачи данных

Выбор коммуникационного оборудования. Ранее были разобраны критерии выбора поставщика коммуникационного оборудования. Здесь мы приведем обзор технических характеристик, на которые стоит обратить внимание. Для простоты будем рассматривать выбор коммутатора 2-го уровня OSI, который может обеспечивать передачу данных со скоростью до 10 Гбит/с.

1. При проектировании необходимо учесть требования к масштабированию сети. Базовая характеристика любого коммутатора — это его пропускная способность, которая, по сути, определяет пропускную способность сети (в случае лишь одного коммутатора). Однако рано или поздно возникнет вопрос о расширении сети. Классический подход к расширению сети подразумевает использование так называемого транка (trunk) — соединения двух коммутаторов между собой, используя свободные порты. Пропускная способность транка не может быть выше пропускной способности порта коммутатора (в случае, если коммутаторы соединены через один порт). Современный метод объединения коммутаторов подразумевает установку коммутаторов в специальный шкаф (rack), оборудованный интерфейсом для подключения дополнительных коммутаторов. Таким образом, расширение сети происходит путём подключения нового коммутатора по принципу дополнительного модуля. Пример такого оборудования представлен на рис. 8.9. Данный подход удобен быстрым расширением пропускной способности сети и отсутствием узкого места в виде транка: интерфейс соединения внутри шкафа работает несоизмеримо быстрее, чем сетевое соединение. Основной и единственный недостаток такого подхода — чрезвычайно высокая стоимость подобного оборудования в сравнении с классическим коммутатором.

2. Далее необходимо решить вопрос соединения коммутатора с выбранной физической средой. Современное сетевое оборудование, как правило, предлагает дополнительный уровень модульности, помимо объединения в группы коммутаторов: использование разных сред передачи в рамках одного коммутатора. Это достигается путём использования дополнительных модулей (рис. 8.10). Основной стандарт на 2014 год — семейство SFP (Small form-factor pluggable transceiver). Наиболее распространённые версии — SFP+ (до 10 Гбит/с) и новейший QSFP (до 40 Гбит/с). Второй обладает несколько иным форм-фактором, однако принадлежит к тому же семейству технологий. Отличие в форме вызвано лишь тем, что QSFP агрегирует 4 SFP+ модуля вместе. Модули SFP предоставляют конечный интерфейс, к которому уже можно подключить выбранную физическую среду.

3. Учтем необходимость реализации концепций SDN в оборудовании. Некоторые производители предлагают решения SDN на про-

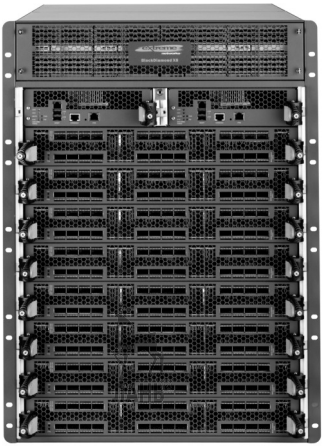


Рис. 8.9. Модульный коммутатор компании Extreme Networks [7]

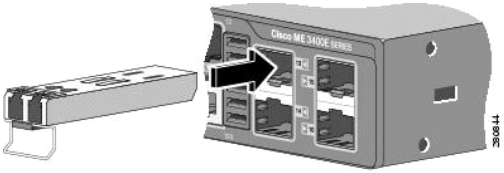


Рис. 8.10. Принцип модульности современного коммутационного оборудования [8]

граммном уровне, некоторые — на аппаратном. Во втором случае производительность данной системы будет несоизмеримо выше, однако может на порядок изменить стоимость оборудования. Стоит также отметить, что программную реализацию SDN большинство производителей предлагает даже для наиболее бюджетных моделей оборудования.

Таким образом, выбор оборудования делается, исходя из трёх независимых факторов:

- требования к масштабированию сети;
- требования к варианту доступа в физической среде;
- необходимости реализации SDN на аппаратном уровне.

Выбор коннектора. Под коннектором подразумевается интерфейс, используемый для подключения физической среды к коммутационному оборудованию. В области высокоскоростных сетей существует два вида соединений:

- SFP-порт сетевого оборудования → SFP-модуль → Кабель с каким-либо коннектором;
- SFP-порт сетевого оборудования → Кабель с прямым подклю-

чением SFP (Direct Attach SFP, DA SFP).

Таким образом, в первом случае имеется необходимость в использовании двух видов компонентов, требуемых для установления соединения между двумя сетевыми устройствами:

– кабель с коннектором;
– два SFP-модуля.

В то же время для второго случая требуется лишь одно устройство — кабель SFP с прямым подключением.

Почему же не ограничиться использованием второго варианта, который значительно более эффективен в плане монтажа, обслуживания, надёжности и стоимости? У DA SFP есть ряд недостатков, основными из которых являются:

– максимальная длина не более 10 м, что часто недостаточно;
– форм-фактор кабеля: он достаточно толстый, что затрудняет прокладку кабеля через каналы коммутационного шкафа (cable management).

На рис. 8.11 представлен типичный кабель DA SFP. Видно, что его габариты значительно превышают габариты стандартного медного кабеля и тем более оптического. При этом он остаётся единственной относительно дешёвой альтернативой классическому варианту построения высокоскоростной сети при условии малого расстояния между сетевыми элементами и невысокой плотности кабелей в канале.

Рис. 8.11. DA SFP кабель [9]

Для первого варианта необходимо приобретать по отдельности как SFP-модуль для коммутатора, так и подходящий кабель. Данный подход устраняет недостатки DA SFP, однако удорожает систему в 2–3 раза.

Сетевые карты устройств. Современная сеть (особенно высокоскоростная) немыслима без вспомогательных устройств, таких как, например, балансировщик нагрузки. Очень часто основой такого устройства является обычный сервер, действующий как шлюз. По этой причине стоит отметить, какие требования предъявляются к подобного рода устройствам при построении высокоскоростной сети.

В наиболее общем случае устройство, такое как балансировщик

нагрузки (БН), принимает пакет и направляет его через шину PCIe на другой интерфейс, действуя по принципу шлюза (gateway). Несложно догадаться, что основным слабым местом такой системы является её сетевая карта (NIC), а именно плата, отвечающая за приём пакета из физической среды и перенаправление его в шину (bus).

Существует множество производителей, предлагающих высокоскоростные сетевые карты. Все они в целом обладают схожей производительностью и функционалом, однако критическим моментом является поддержка той или иной карты на уровне операционной системы. Это является важным фактором для достижения наивысшей производительности и является едва ли не главным критерием выбора NIC. Однако помимо поддержки на уровне OS, существует проблема возможности установки определённого типа NIC в определённую модель сервера. Такие ограничения часто позволяют себе накладывать очень крупные производители серверов, вынуждая тем самым использовать строго определённые NIC, которые были протестированы для работы с аппаратными компонентами определённого производителя. С одной стороны, это приводит к увеличению общей надёжности и стабильности системы, с другой — уменьшает выбор компонентов при покупке оборудования.

Выбор протокола передачи данных. Как было показано ранее, TCP не всегда способен справиться с задачей, передавать данные со скоростью в несколько гигабит в секунду. Однако при небольших расстояниях это возможно. С появлением потерь или джиттера в сети, передача с высокой скоростью с помощью TCP не представляется возможной. Возможны два варианта решения поставленной задачи:

Передача нескольких потоков. В принципе, данный подход является решением поставленной задачи, однако у него есть ряд «подводных камней»:

1. Число потоков одновременной передачи данных имеет смысл повышать только до определённого предела, поскольку, начиная с определенного значения (которое будет варьироваться в основном в зависимости от модели CPU), количество ресурсов, выделяемых системой на поддержку каждого потока, превысит количество ресурсов, выделяемых системой для обработки пользовательского трафика.

2. В современном Интернете информационная безопасность является для многих пользователей приоритетом номер один, особенно в корпоративном секторе. Почти любая корпоративная сеть использует межсетевые экраны (firewall) для контроля безопасности сети. Для классического сетевого экрана наличие множества соединений, исходящих из одного источника и передающих большой объем трафика, является достаточно подозрительным сигналом. Такое поведение мо-

жет даже привести к блокировке источника трафика. Во избежание этой проблемы необходим прямой доступ к сетевому экрану для его конфигурации, что не всегда представляется возможным (например, в случае проблем с сетевым экраном провайдера связи).

Таким образом, для обеспечения возможности высокоскоростной передачи данных в несколько потоков с использованием TCP необходимы два условия:

- наличие чрезвычайно мощного многоядерного процессора, который будет в состоянии обслужить достаточное количество потоков;
- полный доступ ко всей инфраструктуре для обеспечения возможности корректной конфигурации сетевого экрана.

Данные условия могут быть выполнены, однако такие ресурсы доступны не каждому пользователю. Примером реализации подобного подхода является протокол Grid FTP.

Использование альтернативных протоколов на основе UDP. Как было показано ранее, TCP не справляется с задачей высокоскоростной передачи данных в основном из-за проблем с устаревшим дизайном протокола, который разрабатывался десятки лет назад. Поскольку создание принципиально нового протокола является достаточно трудоёмким процессом с последующей необходимостью внедрения на сетевых устройствах производителей (vendors), было принято решение использовать UDP как базу для разработки протоколов нового поколения, решающих проблему высокоскоростной передачи данных. UDP является намного более простым протоколом, чем TCP, и представляет широкое поле для надстроек на уровне приложения. Такой подход имеет ряд преимуществ, основными из которых являются:

- полная поддержка UDP на любом устройстве с сетевым интерфейсом. Таким образом, не возникает проблем с совместимостью на четвертом (транспортном) уровне модели OSI;
- широчайшие возможности для реализации инновационного пользовательского функционала на седьмом уровне модели OSI (уровень приложения).

Возникает вопрос: «Как же можно сравнивать TCP и UDP — протоколы, решающие принципиально разные задачи достоверной и недостоверной доставки информации?». И в самом деле, сам по себе UDP является лишь базисом, транспортной основой для нового вида протоколов. В то время как логику обеспечения достоверной доставки приходиться реализовывать поверх UDP. И это является не единственной проблемой при разработке нового протокола. Помимо достоверности доставки важны такие вопросы, как контроль перегру-

зок (congestion control) и контроль соединения (flow control). Таким образом, имея в качестве базы протокол UDP, необходимо реализовать три модуля для обеспечения полноценной замены TCP:

- модуль организации достоверной доставки (reliability scheme);
- модуль контроля перегрузок (congestion control);
- модуль контроля соединений (flow control).

Учитывая недостатки TCP, описанные ранее, можно предложить более эффективные реализации каждого из этих модулей, добившись тем самым существенного прироста производительности по сравнению с TCP.

Подходя к вопросам доработки UDP до протокола достоверной доставки информации, нельзя не упомянуть важность выбора языка программирования, на котором будет написана новая реализация. Каждый язык программирования решает свою задачу, и при разработке телекоммуникационного протокола необходимо учесть следующие:

- удобный интерфейс работы с оперативной памятью;
- наличие современных библиотек разработки сетевых приложений;
- возможность работать с аппаратной частью машины;
- безопасность работы с оперативной памятью, особенно в вопросах утечки памяти (memory leaks).

На сегодняшний день (2015 год) наиболее явными кандидатами являются языки, представленные в табл. 8.1.

Как показано в табл. 8.1, каждый язык хорош по-своему и выбор сделать непросто, особенно учитывая, что это лишь малая часть языков, подходящих для решения поставленной задачи. Однако грамотная оценка требований к производительности, наличие собственного опыта и сроков разработки помогут сделать правильный выбор.

Примером реализации подобного подхода является UDT.

8.5. Неявные проблемы обеспечения высокоскоростной передачи данных

При работе над созданием протоколов для высокоскоростной передачи данных необходимо также учесть следующие моменты:

- сохранение данных;
- буферизация данных на уровне приложения;
- требования к CPU.

Данные проблемы часто остаются «за кадром», однако они оказывают огромное влияние на качество передачи информации в высокоскоростной сети.

Сохранение данных. При обычной передаче данных проблема обеспечения записи данных на носитель не возникает. Однако представим себе поток данных в 10 Гбит/с, который несёт полезную на-

Таблица 8.1. Сравнение языков программирования для разработки сетевых приложений

Язык	Достоинства	Недостатки
C	Наиболее профессиональное сообщество. Поддержка на большинстве платформ. Лучший интерфейс для работы с аппаратной частью	Отсутствие поддержки многопоточности. Множество ручных операций при работе с памятью
C++ 11	Поддержка многопоточности. Широкие возможности работы с памятью. Совместимости с библиотеками и приложениями языка C. Возможно, лучшие библиотеки разработки сетевых приложений	Относительно высокая сложность программирования. Все еще не совершенная модель работы с памятью
Haskell	Высокая скорость разработки. Активное сообщество. Наиболее безопасное программирование при работе с памятью	Высокая сложность программирования
RUST	Простота разработки. Активное сообщество	Не такая широкая поддержка, как у конкурентов
Java	Полная независимость от платформы — один код можно использовать на разных системах. Множество сторонних библиотек	Использование виртуальной машины для запуска наносит вред производительности

грузку, необходимую пользователю. Процесс получения выглядит следующим образом:

- пакет приходит на высокоскоростную NIC (сетевую карту);
- отбрасывается IP-заголовок;
- пользовательские данные передаются в шину;
- далее жёсткий диск должен записать данные.

На последнем этапе возникает проблема: поток данных в 10 Гбит/с соответствует скорости записи на диск в 1,16 Гбайт в секунду! Но не существует диска, способного записать данные с подобной скоростью. Даже самые современные твердотельные накопители (Solid State Drive, SSD) не позволят записать данные со скоростью быстрее 700 Мбайт в секунду. Таким образом, каждую секунду очередь на запись будет расти более чем на 400 Мбайт, что приведёт к

скорейшему переполнению буфера.

У данной проблемы существует два решения:

1. Запись данных изначально производится в оперативную память с последующим копированием их на диск. Такой подход применим только при наличии огромного объема оперативной памяти, в том числе свободной, что часто проблематично.

2. Использование массивов дисков как RAID или SAS, которые позволят добиться высокой скорости записи путём распараллеливания записи. Такой подход требует покупки достаточно дорогих контроллеров, однако является единственным доступным решением на сегодняшний день.

Буферизация данных на уровне приложения. Данная проблема также вызвана огромным количеством пакетов, которые необходимо сгенерировать при передаче высокоскоростного трафика. Кроме того, реализация собственной схемы достоверной доставки, описанная ранее, накладывает требование на наличие пользовательского буфера в приложении. Существующие реализации для буферизации данных часто не позволяют добиться требуемой производительности. И именно необходимостью реализации собственного буфера обусловлены высокие требования к интерфейсу работы с памятью у выбранного языка программирования. Правильная реализация пользовательского буфера решает следующие фундаментальные задачи:

- сохранения большого количества пользовательских данных, а именно данных, ожидающих подтверждения от получателя;
- сглаживание эффектов вариации сетевой задержки (джиттера), которые часто являются губительными для протоколов передачи данных, вызывая перемешивание (reordering) пакетов в сети.

Таким образом, пользовательский буфер является неотъемлемой частью высокоскоростного протокола передачи данных. Стоит заметить, что система обеспечивает каждое сетевое соединение внутренним буфером, который выполняет несколько иные задачи, а именно сохранение пакетов, ещё не переданных на уровень приложения. И при организации высокоскоростной передачи данных размер этого буфера должен быть значительно расширен (каждая система предлагает для этого свой интерфейс) в связи со значительно возросшей интенсивностью прибывающих пакетов.

Использование CPU. Давно уже имеет место стереотип: «чем больше ядер процессора, тем лучше». Однако это верно лишь отчасти. Да, существуют профессиональные приложения, оптимизированные для работы в многоядерной среде, и использование там многоядерных процессоров более чем целесообразно. Но возвращаясь к

тематике высокоскоростной передачи данных, заметим, что подавляющее большинство систем основаны на системе Linux, ядро которой написано на языке C, который не имеет поддержки многопоточности (не может разделять выполнение задач между несколькими потоками выполнения). Таким образом, обработка пакетов, приходящих с NIC, так или иначе будет осуществляться в один поток, что в свою очередь означает нецелесообразность применения большого числа ядер в системах высокоскоростной передачи данных. Данное утверждение верно до тех пор, пока существующее ядро Linux господствует на рынке телекоммуникаций. Применение одноядерных систем тоже не целесообразно, поскольку незадействованные вычислительные ресурсы могут понадобиться при работе на уровне приложения.

Итак, следует подчеркнуть, что имеет смысл выбирать системы с более высокой тактовой частотой ядер, а не гнаться за их количеством.

В заключение заметим, что ряд поставленных авторами в главе вопросов и ответов на них может кому-то из читателей показаться спорными, по некоторым вопросам у них могут появиться свои предложения. В этом-то и заключалось использование этого материала в учебном пособии, в котором, по мнению авторов, должны не только излагаться давно известные факты, но и по возможности приводиться перспективные исследования, которые будут воплощены в жизнь в ближайшие годы.

8.6. Список литературы

1. *Paxson V.* End-to-End Internet Packet Dynamics // IEEE/ACM Transactions on Networking, June 1999. — P.277–292.
2. *Wang Y.A., Huang C., Li J., Ross K.W.* Queen: Estimating Packet Loss Rate between Arbitrary Internet Hosts // Proceedings of the 10th International Conference on Passive and Active Network Measurement, 2009. – P.57–66.
3. *Settlemyer B.W., Rao N.S.V., Poole S.W., Hodson S.W., Hick S.E., Newman P.M.* Experimental analysis of 10 Gbps transfersover physical and emulated dedicated connections // Proc. of Computing, Networking and Communications (ICNC). Maui, Hawaii, USA, 2012. — P.845–890.
4. *Kachan D., Siemens E., Shuvalov V.* Comparison of contemporary solutions for high speed data transport on WAN connections // Proceedings of ICNS 2013, Lisbon. — P.34–43.
5. *Kachan D., Siemens E.* Comparison of Contemporary Protocols for High-speed Data Transport via 10 Gbps WAN Connections // Proceedings of the 2nd International Conference on Applied

Innovations in IT, 2014. — P.21–27.

6. *Yunhong Gu, Xinwei Hong, Mazzucco M., Grossman R.* SABUL: A High Performance Data Transfer Protocol // Journal of Grid Computing, 2004. — P.377–386.
7. http://assets.extremenetworks.com/sites/default/files/null/BlackDiamondX8_Top_Front.jpg
8. http://www.cisco.com/c/dam/en/us/td/i/200001-300000/280001-290000/280001-281000/280844.eps/_jcr_content/renditions/280844.jpg
9. http://h10003.www1.hp.com/digmedialib/prodimg/lowres/c01824150.jpg

Приложение 1

Программно-конфигурируемые сети

П.1. Общие положения

С каждым годом, начиная с 2006 г., появляется все больше и больше работ, посвященных программно-конфигурируемым сетям (ПКС) или SDN (Soft Defined Networks), предлагаются программные продукты, техника, позволяющие реализовать эти сети. В этом разделе мы попытаемся, используя известные публикации [1–14], рассказать, что такое ПКС, почему вдруг появилась необходимость в новой технологии построения компьютерных сетей, каковы перспективы внедрения этой технологии в мире, место России в продвижении этой технологии.

Что не так в современных IP-сетях? Появление Интернета было революционным событием, изменившим нашу экономическую, технологическую и социальную жизнь. Несмотря на то, что его общая архитектура несомненно является успешной, современный Интернет достаточно дорогой и сложный в управлении, слишком завязанный на вендеров и не гибкий для развития. В такой сети при появлении оборудования нового поставщика (вендора) оператору необходимо модернизировать всю систему управления сетью.

Основные сетевые протоколы TCP/IP были разработаны в 70-е годы прошлого века на заре Интернета. С тех пор по мере необходимости предоставления все новых и новых услуг к этим протоколам добавлялись все новые и новые протоколы. Сегодня количество протоколов превышает 600 и продолжает расти. Сеть становится все сложнее и соответственно усложняется управление этой сетью.

В сети традиционной архитектуры технологии виртуализации вычислительного окружения и хранилищ данных прочно закрепились в дата-центрах, отлажены и хорошо работают. Но в сетевом окружении, как раньше, так и сейчас существуют разные трудности [1]:

- статическое или ручное выделение и перераспределение сетевых ресурсов;
- отдельная настройка каждого сетевого устройства при большом их количестве;
- сложность и ресурсоемкость при внедрении и изменении сетевых политик, конфигураций, новых сервисов;
- многовендорность и проприетарность некоторых функций;

В классической IP-сети маршрутизатор содержит две плоскости: плоскость управления (control plane) и плоскость данных (data plane). Плоскость управления в маршрутизаторе обрабатывает пакет и при-

нимает решение, куда его передавать дальше (операция «routing»). В плоскости передачи данных решается проблема продвижения пакета от входного порта на определенный выходной — операция, которая называется в англоязычной литературе «forwarding». Все эти операции определяются заложенными в маршрутизатор конкретными протоколами, указаниями, что надо «делать то», если «имеем это». Что-либо менять в этом маршрутизаторе, если появится необходимость в новой услуге, требующей других алгоритмов работы, далеко не всегда получится без смены оборудования (железа), в котором будут выполняться другие алгоритмы.

Многие крупные вендоры, в том числе IBM, видят решение вышеописанных задач в использовании программно-конфигурируемых сетей, которые полностью изменят экономику и опыт внедрения IT-систем [1].

Что предлагает технология ПКС:

- отделить в маршрутизаторе управление сетевым оборудованием от управления передачей данных, а управление вынести на отдельный компьютер, который будет находиться под контролем администратора сети;
- перейти от управления отдельным экземпляром сетевого оборудования к управлению сетью в целом;
- создать интеллектуальный программно-управляемый интерфейс между сетевым приложением и транспортной средой.

Еще раз напомним, что в классической сети задача построения маршрута (control plane) и реализация маршрута (data plane) объединены в сетевом устройстве. В SDN предусматривается передача управляющих функций так называемому контроллеру, что приводит к замене традиционной распределенной модели маршрутизации централизованной моделью. Таким образом, процесс управления сетью, включающий создание маршрутов, становится программированием сети в целом.

На рис. П.1.1 представлена классическая сетевая архитектура, состоящая из маршрутизаторов, каждый из которых имеет в своем составе уровень управления (App и OS) и уровень данных (Forwarding).

На рис. П.1.2 представлена ПКС (SDN), в которой управление централизовано. Здесь связь контроллера (Controller) с переключателем Switch осуществляется при помощи протокола OpenFlow. Приложения (Application, App) и контроллер общаются через графический интерфейс пользователя (Graphical User Interface, GUI).

На рис. П.1.1 можно выделить три уровня: уровень приложений (App), уровень управления (Controller) и уровень инфраструктуры сети (Switch). Для управления инфраструктурой сети необходим сбор

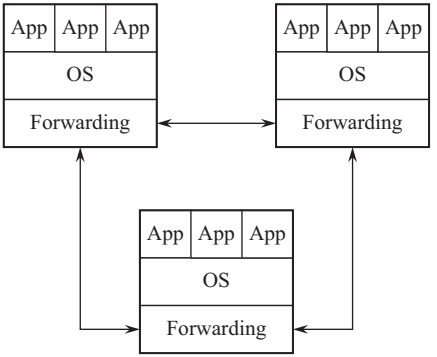


Рис. П.1.1. Классическая сетевая архитектура

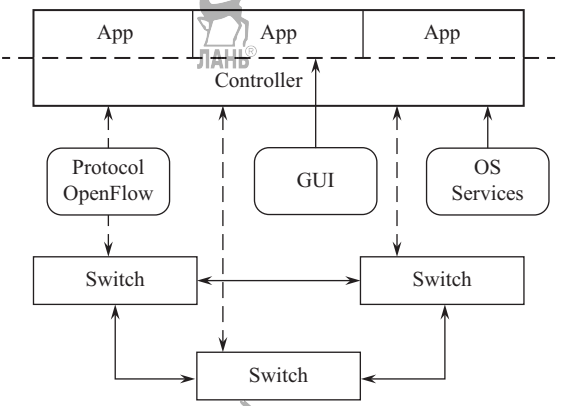


Рис. П.1.2. Программно-конфигурируемая сеть

данных о состоянии сети. Объем этих данных чрезвычайно велик и их можно отнести к категории Big Data (большие данные). Обработка этих данных является сложной задачей, относящейся к категории NP-Complete: необходимо оценить степень удовлетворенности услугами пользователей; измерить сетевые параметры, характеризующие QoS; определить топологию и междоменное взаимодействие; составить прогноз изменений в топологии и сетевых характеристик и т.д. В этой связи в [2] предлагается в SDN добавить четвертый уровень или четвертую информационную плоскость (рис. П.1.3)

Переход на технологию ПКС упрощает процесс создания маршрутов, увеличиваются возможности для инноваций, введение которых осуществляется написанием нового приложения (перепрограммированием). И, наконец, вместо сложных и дорогостоящих маршрутизаторов можно использовать более простые устройства.

Отметим, что разделение уровня управления и уровня переда-

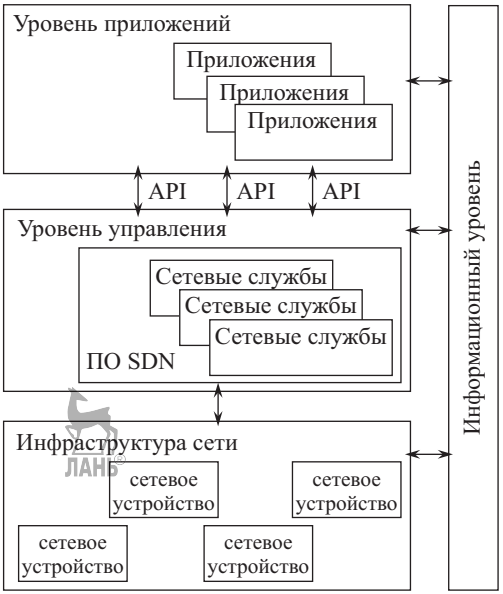


Рис. П.1.3. Четырехуровневая архитектура SDN

чи данных — это общая идея (кстати, далеко не новая — вспомним Softswitch), имеющая множество способов реализации. Так, можно использовать для управления один контроллер, а можно несколько. У каждого из этих вариантов есть свои плюсы и минусы. Можно полностью убрать функции управления из маршрутизации, оставив только функцию продвижения, а можно часть функций управления оставить за маршрутизатором. Понятно, что централизация управления (использование на всю сеть одного контроллера) скажется на структурной надежности сети. Децентрализованная система управления сложна с точки зрения реализации, но более живучая при отказах. Использование «тупых» и, следовательно, дешевых маршрутизаторов на сети может привести к полному параличу на сети при выходе из строя контроллера.

Технический директор Cisco в России и СНГ Андрей Кузьмич, говоря о характерных особенностях SDN, отмечает, что сила SDN не просто в вынесении таблицы коммутации/маршрутизации в некий контроллер, а в развитии дополнительных сервисов как отдельных программных продуктов, которые будут влиять на построение и изменение таблиц коммутации/маршрутизации. Например, исходя из важности конкретного приложения, контроллер сможет менять таблицу коммутации для его приоритетного обслуживания [3].

Следует заметить, что технология ПКС — это развивающаяся

технология, концепция которой зародилась в Стэнфордском университете, исследовательской группе которого потребовалось создать тестовую среду для экспериментов с новыми сетевыми протоколами. Строить отдельную сеть было дорого, поэтому решили задействовать имеющуюся университетскую сеть, в которой с помощью прообраза SDN были выделены ресурсы для испытаний. Интерес к SDN со стороны неуниверситетских кругов первыми проявили крупные поставщики интернет-сервисов, которым требовались высокопроизводительные инфраструктуры для организации взаимодействия между десятками и даже сотнями серверов в гигантских ЦОД.

П.2. Протокол OpenFlow и OpenFlow-коммутатор

Наиболее перспективным и активно развивающимся стандартом для SDN является OpenFlow — открытый стандарт, в котором описываются требования, предъявляемые к коммутатору, поддерживающему протокол OpenFlow для удаленного управления.

Openflow — протокол управления процессом обработки данных, передающихся по сети передачи данных, обеспечивающий связь между уровнем управления и устройствами продвижения (передачи) данных (маршрутизаторами и коммутаторами).

Как и следует из названия, протокол OpenFlow при идентификации трафика оперирует понятием «потока», то есть управление данными в OpenFlow осуществляется не на уровне отдельных пакетов, а на уровне их потоков. Правила в коммутаторе OpenFlow устанавливается с участием контроллера только для первого пакета, а затем все остальные пакеты потока его используют.

Ключевым элементом коммутатора, поддерживающего этот протокол, является таблица потоков (flow tables). Протокол используется для управления (программирования) таблиц потоков (flow tables) сетевых коммутаторов и маршрутизаторов с центрального устройства — контроллера сети (сервера или даже персонального компьютера). Коммутаторы с поддержкой OpenFlow могут быть двух типов: OpenFlow-only (работающий только под управлением OpenFlow) или OpenFlow-hybrid (гибридные, совмещающие openflow и обычную обработку пакетов средствами микропрограммы устройства).

Протокол OpenFlow — открытый стандарт, поэтому использовать его в своих устройствах может любой производитель телекоммуникационного оборудования. Уже сейчас этот протокол поддерживают многие производители Ethernet-коммутаторов. Первыми это стали делать NEC и BigSwitch, чуть позже — Pronto и Marvell. Затем к ним присоединились IBM, Brocade и HP, и только после этого о поддержке OpenFlow заявили лидеры рынка IP-коммутаторов — Cisco, Juniper и Extreme Networks [4].

В настоящий момент протокол имеет версию 1.5 (принята 19 декабря 2014 года).

Таблица потоков (flow tables). Ключевым элементом коммутатора, поддерживающего этот протокол, является таблица потоков. Каждая запись в таблице потоков имеет три поля: заголовок PDU (фрагмента данных), который позволяет определить соответствие PDU потоку, действие и поле со статистикой (число байтов и PDU, соответствующее потоку, время прохождения последнего соответствующего потоку PDU).

Заголовок может состоять из множества полей разного уровня (например, MAC-адресов отправителя и получателя, полей из заголовка IP-пакета, полей из заголовка TCP-сегмента), — см. рис. П.1.4, рис. П.1.5.

Каждый логический коммутатор OpenFlow содержит одну или несколько таблиц потоков (рис. П.1.6).

Коммутатор SDN. Согласно спецификации OpenFlow, каждый коммутатор состоит из следующих компонентов [5]:

- одной или нескольких таблиц потоков (flow table);
- безопасного канала (secure channel), использующегося для управления коммутатором внешним «интеллектуальным» устройством (контроллером) при помощи OpenFlow;
- поддержки протокола OpenFlow protocol, использующегося для управления. Использование этого протокола позволяет избежать необходимости писать программу для управляемого устройства.

Основные функции контроллера включают в себя определение активных коммутаторов в сети, определение активных портов на коммутаторах, связь с коммутаторами, описания логики коммутации и маршрутизации пакетов по всей сети для заполнения таблиц коммутатора. Контроллер SDN может передавать команды коммутатору или нескольким коммутаторам SDN (одновременно) на добавление, изменение и удаление записей в таблицах.

Порядок работы коммутатора SDN показан на рис. П.1.7 и состоит из следующих шагов [6]:

1. Первый пакет нового потока прибывает на входящий порт коммутатора.

2. Коммутатор SDN проверяет наличие записей в таблицах потоков, соответствующих пакету. Если соответствующая запись найдена, то выполняется шаг 5.

3. Если записи в таблицах потоков не найдены, пакет может быть передан контроллеру SDN по защищенному каналу связи.

4. Согласно алгоритму маршрутизации, контроллер SDN

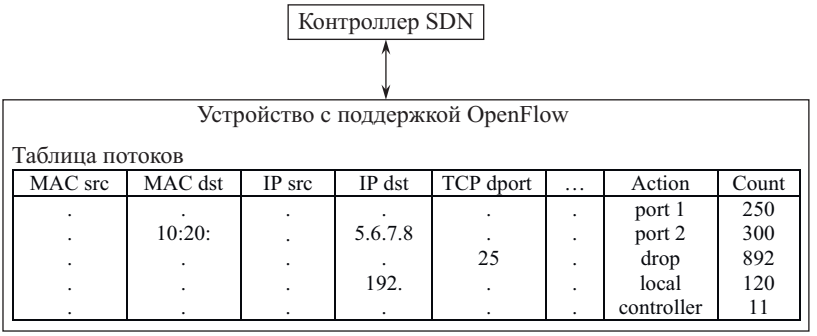


Таблица потоков

MAC src	MAC dst	IP src	IP dst	TCP dport	…	Action	Count
.	.	.	.	.	.	port 1	250
.	10:20:	.	5.6.7.8	.	.	port 2	300
.	.	.	.	25	.	drop	892
.	.	.	192.	.	.	local	120
.	.	.	.	.	.	controller	11

Рис. П.1.4. Типичная таблица потоков в сетевом устройстве, поддерживающем OpenFlow

Рис. П.1.5. Таблица потока OpenFlow

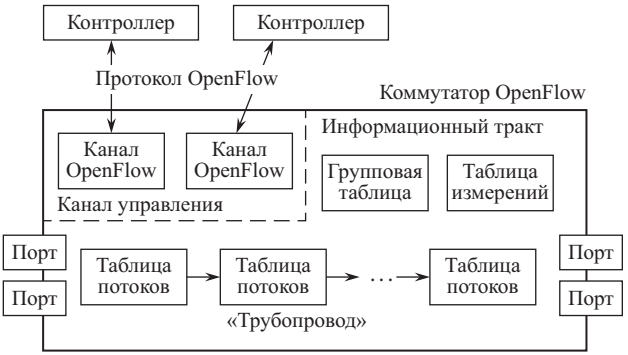


Рис. П.1.6. Основные компоненты коммутатора OpenFlow [3]

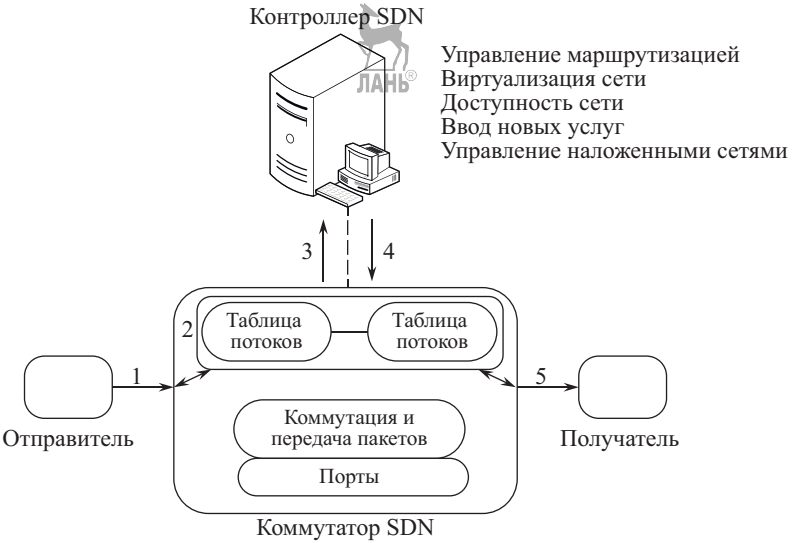


Рис. П.1.7. Порядок работы коммутатора SDN

добавляет соответствующую запись в коммутатор и остальные коммутаторы по тракту передачи данного потока.

5. Коммутатор выполняет инструкции, связанные с данным пакетом, и передает пакет на указанный исходящий порт для дальнейшей его отправки получателю.

Между контроллером и коммутатором возможны три типа сообщений [7]:

- сообщения, инициируемые контроллером, используются для прямого управления коммутатором и для получения информации о его состоянии;
- сообщения, инициируемые коммутатором, используются для извещения контроллера о событиях в сети и изменениях состояния коммутатора;
- сообщения, инициируемые как контроллером, так и коммутатором.

Сообщения, инициируемые контроллером, могут возникать в следующих случаях:

- для получения данных о коммутаторе, его возможностях, его текущей конфигурации;
- для изменения конфигурации, добавления, изменения и удаления записей в таблицах потоков и групп, а также свойств портов.

Сообщения, инициируемые коммутатором, могут возникать в следующих случаях:

– для извещения контроллера об изменениях в состоянии коммутатора SDN;
– для отправки контроллеру SDN полученных пакетов;
– для сообщения об удалении записи из таблицы потоков, изменениях конфигурации и состояния портов, а также о возникающих ошибках.

Сообщения, инициируемые как контроллером, так и коммутатором, могут возникать в следующих случаях:

– для проверки состояния канала связи между контроллером и коммутатором;
– при (начале) установлении соединения между контроллером и коммутаторам;
– для обмена сообщениями об ошибках.

В докладе [7] предлагаются модели массового обслуживания сложной структуры, позволяющие находить основные вероятностно-временные характеристики процессов взаимодействия коммутаторов и контроллера в сети SDN.

П.3. Виртуализация сетей NFV

Идея виртуализации сетевых функций (Network Function Virtualization, NFV), в отличие от SDN, созданного исследователями и разработчиками центров обработки данных (ЦОД), изначально продвигавшаяся крупными европейскими операторами, предполагала перенос сетевых сервисов со специализированных устройств на стандартные компьютерные платформы в виртуализированные среды [9] (рис. П.1.8).

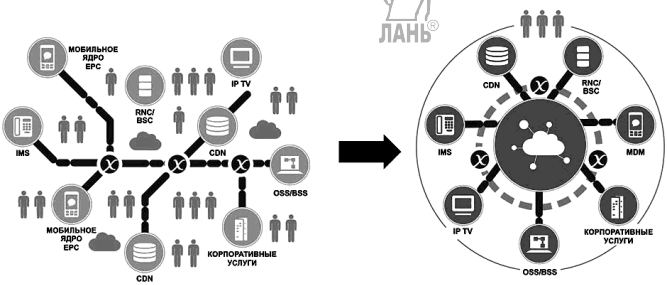


Рис. П.1.8. Принцип NFV

После ряда неудачных попыток сервис-провайдеров увеличить скорость внедрения новых услуг пришло понимание того, что аппаратное решение не позволяет это сделать. Более того, услуги, зависимые от оборудования быстро устаревают, требуя повторения цикла «закупка–планирование–внедрение–использование» с небольшой или

вообще без прибыли. В октябре 2012 г. группа по промышленной спецификации NFV представила официальный документ на конференции, посвященной SDN и OpenFlow в Германии [8]. Сегодня работой по стандартизации NFV занимается европейский институт ETSI (European Telecommunications Standards Institute).

NFV позволяет провайдерам преобразовать сеть с фиксированной, закрытой и зависящей от оборудования конкретного поставщика инфраструктурой в открытую, масштабируемую и адаптируемую для нужных услуг среду. Технология NFV используется в маршрутизаторах, межсетевых экранах и шлюзах, устройствах CDN, акселераторах WAN, контроллерах доставки приложений (ADC) в сетях операторов и сервис-провайдеров.

NFV не потребует отказа от уже развернутой сетевой инфраструктуры при развертывании новых сервисов, так что «новое» может мирно сосуществовать со «старым». Таким образом, внедрить NFV проще, чем SDN, поскольку в NFV используется стандартная аппаратная среда. Кроме того, NFV позволяет вернуться при необходимости к прежней сетевой инфраструктуре, а это снижает риски.

Сегодня основными направлениями NFV являются [10]:

- виртуальные соединения (Virtual Switching) — физические порты, соединенные с виртуальными портами на виртуальном сервере с виртуальными маршрутизаторами, используя виртуализированные IPsec и SSL VPN шлюзы;
- виртуализированные сетевые устройства (Virtualized Network Appliances). Сегодня сетевым функциям необходимы отдельные устройства, которые могут быть заменены их виртуальными аналогами. Например, функции защиты, такие как межсетевые экраны (firewalls) и шлюзы, маршрутизатор широкополосного удалённого доступа (BRAS) и ядро пакетной сети LTE (EPC);
- виртуализированные сетевые услуги (Virtualized Network Services) — например, приложения по управлению сетью, такие как анализаторы трафика и оборудование по мониторингу сети, перераспределители нагрузки и ускорители;
- виртуализированные приложения (Virtualized Applications). Также любые приложения могут быть виртуализированы. Например, важным направлением является развитие облачных приложений, такие как виртуализированные хранилища и услуги оцифровки изображений для соответствия потребностям быстрого роста использования планшетов и смартфонов.

По мнению экспертов, SDN и NFV не связаны между собой, но хорошо дополняют друг друга. Стратегия программно-конфигу-

рируемой сети позволяет провайдерам разорвать зависимость сетевых функций, таких как кэширование, от специализированного оборудования и передать их осуществление виртуализированным приложениям в облачной инфраструктуре, расширяя возможности предоставления услуг [9].

Сравнение SDN и NFV. Сейчас давайте вернемся к взаимосвязи SDN и NFV. Как показано на рис. П.1.9, NFV дополняет SDN, но не зависит от SDN (и наоборот). NFV может быть реализована без SDN, хотя эти два решения могут быть объединены, и тогда потенциальная эффективность будет увеличена [11].

Рис. П.1.9. Соотношение между NFV и SDN

Цель NFV может быть достигнута при использовании не только механизмов SDN, но и опираясь на методы, используемые сейчас во многих ЦОД. Однако подходы, опирающиеся на разделение плоскости управления и передачи, предлагаемые SDN, могут улучшить КПД, упростить совместимость с существующими системами и облегчить процедуры эксплуатации и технического обслуживания. NFV способны помочь SDN, предоставляя инфраструктуру, на которой можно запустить программное обеспечение SDN. Кроме того, NFV развивается в похожем направлении, что и SDN, предполагая использовать сервера и коммутаторы.

Давайте рассмотрим пример того, как SDN и NFV могут работать вместе. На рис. П.1.10 показано, как сегодня осуществляется маршрутизация за счет использования маршрутизатора на стороне клиента. В данной ситуации NFV могло бы применяться для виртуализации функций маршрутизации, как показано на рис. П.1.11. Все что остается на стороне клиента — это сетевое устройство (NID) для сопряжения, а также для оценки производительности.

И, наконец, SDN предлагает разделение плоскости управления и передачи, как показано на рис. П.1.12. Сейчас пакеты данных пере-

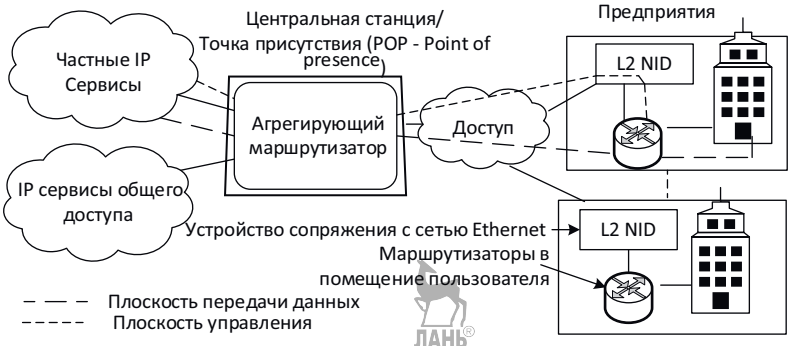


Рис. П.1.10. Управление маршрутизацией сегодня

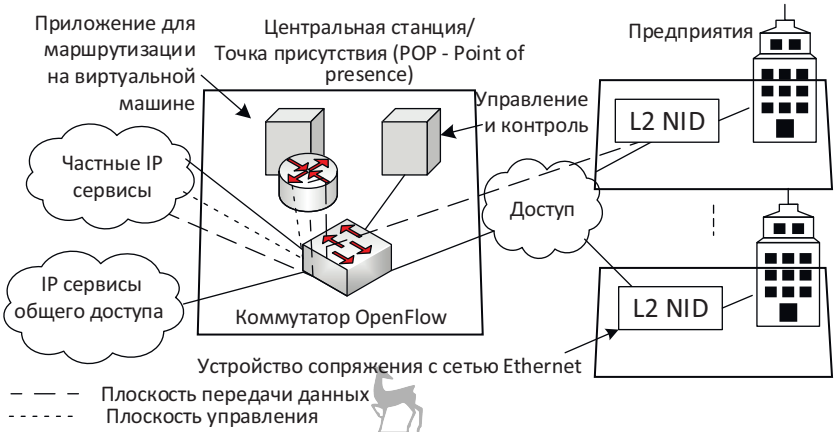


Рис. П.1.11. Управление маршрутизацией с использование NFV

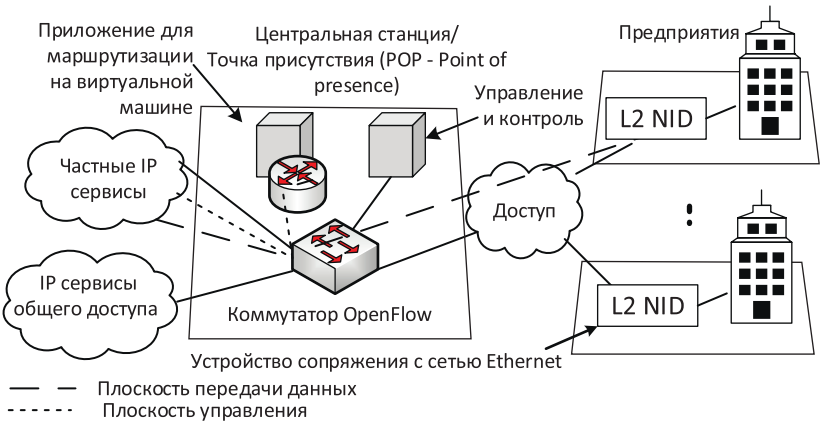


Рис. П.1.12. Управление маршрутизацией с использованием NFV и SDN

Таблица П.1.1. Сравнение SDN и NFV

Категория	SDN	NFV
Причина существования	Разделение управления и передачи, централизация управления и программирования сети	Перемещение сетевых функций от отдельных устройств в сервера
Предполагаемое использование	Кампусы, ЦОД/облако	Поставщик сетевых услуг
Используемые устройства	Сервера и коммутаторы	Сервера и коммутаторы
Изначальное применение	Управление облаками и сетью	Маршрутизаторы, системы сетевой защиты (firewalls), шлюзы, корпоративные сети обмена данными (CDN), ускорители сетей WAN, гарантия SLA (Соглашение об уровне предоставления услуги)
Новые протоколы	OpenFlow	Пока не разработаны
Стандартизация	Открытый сетевой форум (ONF)	Рабочая группа ETSI NFV

даются за счет плоскости передачи данных, в то время, как функции маршрутизации (плоскость управления) запущены на виртуальной машине сервера.

Объединение SDN и NFV, показанное на рис. П.1.12, обеспечивает оптимальное решение:

- дорогое и специализированное оборудование заменяется простым аппаратным и продвинутым программным обеспечением;
- программная плоскость управления перемещается из дорогого оборудования (отдельных платформ) в оптимизированную область (Сервер ЦОД или центр коммутации сети-POP);
- плоскость управления и передачи были отделены и абстрагированы, позволяя сети и приложениям развиваться без необходимости модернизации сетевого оборудования.

В табл. П.1.1 представлены итоговые результаты сравнения SDN и NFV.

Как видно, SDN и NFV не связаны между собой, но хорошо дополняют друг друга. Стратегия программно-конфигурируемой сети позволяет провайдерам разорвать зависимость сетевых функций, таких как кэширование, от специализированного оборудования и передать их осуществление виртуализированным приложениям в облачной инфраструктуре, расширяя возможности предоставления услуг.

П.4. Стандартизация ПКС

Общее положение. Исследованием общих вопросов и стандартизацией SDN занимаются ONF, IETF, Исследовательская группа интернет-технологий (Internet Research Task Force, IRTF), Европейский институт по стандартизации в области телекоммуникаций (European Telecommunications Standards Institute, ETSI) и МСЭ-Т. Представители Форума широкополосных сетей (Broadband Forum, BBF) изучают некоторые частные вопросы построения и эксплуатации сетей SDN. В рамках рабочей группы «Инновационные услуги и рыночные требования» (Service Innovation & Market Requirements) выделен проект SD-313 — коммерческие требования и структура SDN в широкополосных телекоммуникационных сетях (Business Requirements and Framework for SDN in Telecommunication Broadband Networks). Проект предполагает проведение исследований в области сценариев перехода к сетям SDN, включая варианты поддержки SDN частью оборудования, а также внедрение функциональности SDN в оборудование при обновлении ПО. Частные вопросы применительно к SDN рассматриваются и участниками Форума оптического межсетевого взаимодействия (Optical Internetworking Forum, OIF). Эта некоммерческая организация, разрабатывающая соглашения по реализации (Implementation Agreement, IA) для оборудования оптических сетей, оценивает концепцию SDN как перспективную и занимается разработкой требований к SDN в части транспортных сетей со стороны операторов (оптических) сетей и поставщиков услуг, структуры SDN и ее соотносимости с архитектурой оптических сетей с автоматической коммутацией (Automatically Switched Optical Network, ASON), а также демонстрацией и тестированием SDN [12].

Стандарты ONF. В 2011 г. компании Facebook, Deutsche Telekom, Microsoft, Verizon и Yahoo! организовали консорциум ONF с целью развития технологий SDN в целом и протокола OpenFlow в частности. Сегодня членами ONF являются практически все основные поставщики сетевого оборудования, включая Alcatel-Lucent, Brocade, Ciena, Cisco, Dell, Ericsson, Extreme Networks, HP, Huawei, IBM, Infinera, Intel, Juniper Networks, Mellanox, Netgear, Nokia Solutions and Networks, ZTE, а также лидеры рынка систем виртуализации VMware и Citrix. Основной задачей ONF является представление стандарта OpenFlow, который позволяет осуществлять удаленное управление уровнем передачи данных. Стандарт OpenFlow, первый стандарт SDN, является существенным элементом в открытой архитектуре SDN. В настоящее время в ONF продолжается работа по анализу требований к SDN, развитию стандарта OpenFlow в соответствии с запросами, возникающими при коммерческом развертывании

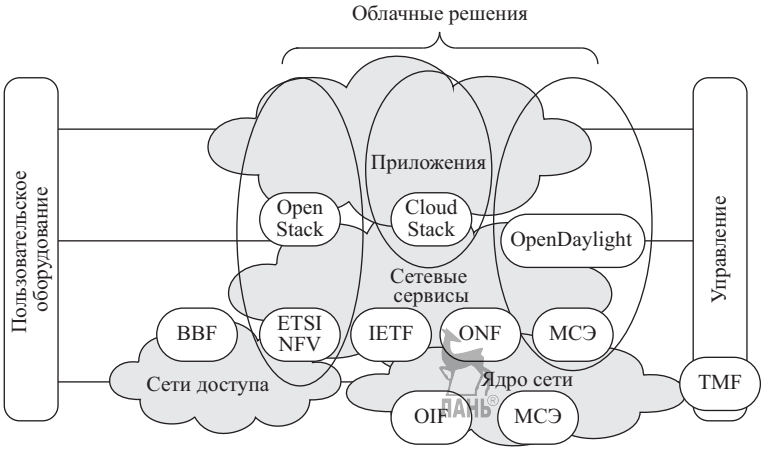


Рис. П.1.13. Распределение стандартизирующих организаций по направлениям разработки архитектуры SDN

SDN, а кроме того, создаются новые стандарты в целях расширения возможностей SDN.

Стандарты IETF. Работа IETF, сообщества ученых, операторов сетей связи и производителей сетевого оборудования, исследующих вопросы развития сети Интернет и ее функционирования, распределена между РГ согласно основным выделенным тематикам, таким как маршрутизация, транспорт, безопасность и пр. Разработка аналитических и стандартизирующих документов IETF, относящихся к SDN, стартовала в конце 2012 г. и сейчас находится на начальных этапах.

Стандарты МСЭ-Т. МСЭ-Т занимается исследованием технических, эксплуатационных, тарифных и других вопросов, выпускает рекомендации с целью стандартизации электросвязи на международном уровне. В ходе Всемирной ассамблеи МСЭ по стандартизации электросвязи в Дубае в 2012 г., где обсуждались вопросы SDN, было высказано согласованное мнение, что программно-конфигурируемые сети коренным образом преобразуют отрасль электросвязи и ИКТ в ближайшие десятилетия, обеспечат отрасли многочисленные преимущества. В условиях быстро растущего интереса к SDN со стороны значительного количества компаний необходима система стандартов для широкого применения SDN. По результатам обсуждения тематики SDN в Исследовательской комиссии (ИК) 13 МСЭ-Т было поручено в следующем исследовательском периоде расширить и ускорить работы в области архитектуры и требований к SDN, а также представить рекомендации Консультативной группе по стандартизации электросвязи (КГСЭ) относительно порядка рассмотрения вопросов, выходящих за рамки мандата ИК 13. КГСЭ поручено изучить проблему,

рассмотреть вклады ИК 13 и других ИК и принять необходимые меры для активизации деятельности по стандартизации SDN в МСЭ-Т [13]:

- определить соответствующие ИК для осуществления последующих действий и установить подходящую организационную структуру в отношении SDN;
- координировать работу по техническим вопросам SDN между ИК в соответствии с их компетенцией;
- содействовать развитию сотрудничества с другими органами и форумами по стандартизации, занимающимися вопросами SDN;
- определить четкое стратегическое видение процесса стандартизации SDN и активную роль МСЭ-Т.

В настоящее время исследованиями в области SDN в МСЭ-Т занимаются ИК 13 (архитектуры и функциональные требования к SDN) и ИК 11 (эталонные архитектуры сигнализации SDN, требования к сигнализации и протоколам SDN, включая протоколы взаимодействия, а также тестирование на соответствие и взаимодействие). Интерес к SDN проявляют ИК 15 (транспорт в SDN) и ИК 17 (безопасность в SDN) [12].

Стандарты ETSI. Институт ETSI — некоммерческая организация по разработке стандартов в области телекоммуникаций, официально признанная Европейским союзом. Институт разрабатывает стандарты фиксированной, подвижной, радио, конвергентной связи, а также телевещания и интернет-технологий. В составе ETSI была выделена группа отраслевой спецификации (Industry Specification Group, ISG) по разработке концепции виртуализации сетевых функций (служб) (Network Functions Virtualisation, NFV) [19]. Концепция NFV предполагает замещение разнообразных сетевых устройств стандартизированными высокопроизводительными серверами, коммутаторами и системами хранения данных с реализацией сетевых функций (служб) программным обеспечением.

Предполагается, что ISG NFV разработает требования и архитектуру NFV, рассмотрит вопросы управления, оркестровки служб, архитектуры ПО, производительности и переносимости, надежности и устойчивости, безопасности и миграции к NFV. Концепция NFV сходна с концепцией SDN, однако на текущем, начальном, этапе работы степень их взаимного соотношения не является точно определенной. Согласно ETSI, концепция NFV в большой степени дополняет SDN, но концепции независимы, каждая из них может быть реализована отдельно. Сегодня стандарты группы ISG NFV находятся на начальной стадии разработки. В октябре 2013 г. группой были разработаны первые спецификации [12].

П.5. SDN в России

Компания J'son & Partners Consulting представила основные результаты опроса крупнейших российских телекоммуникационных операторов о планах внедрения технологий SDN и NFV на коммерческих сетях.

По результатам опроса J'son & Partners Consulting, по состоянию на июль 2014 г. большая часть опрошенных операторов (более 70%) мобильной и фиксированной связи находилась на стадии изучения и анализа возможностей SDN и NFV. На этапе тестирования технологий — более половины операторов. Подавляющее большинство опрошенных российских операторов считают, что внедрение технологий SDN/NFV на коммерческой сети состоится в ближайшие 2–3 года, т.е. в 2016–2017 гг. Около 7% респондентов считают, что это произойдет уже в 2015 г.

Более 70% опрошенных операторов в качестве одного из главных драйверов развития SDN/NFV называют сокращение сроков модернизации сети за счет упрощения сетевой инфраструктуры. Почти 2/3 опрошенных операторов в тройку основных драйверов развития SDN и NFV отнесли снижение OPEX.

По состоянию на сентябрь 2014 г. российские операторы планируют или уже проводят тестирование SDN на своих сетях. В оптимистическом сценарии первые внедрения на коммерческой сети можно ожидать в течение ближайшего года, более реалистичные сроки — 2–3 года. При этом концепция NFV более «сырая», операторы выжидают успешных кейсов на мировом рынке. Напомним, однако, что еще в начале года об NFV говорилось как об очень перспективной сетевой технологии.

Недостаточная информированность о технологиях является самым главным барьером для внедрения SDN/NFV. К другим сдерживающим факторам, выделяемым операторами, относятся, например, неподтвержденность экономической эффективности, неготовность вендоров и риски снижения надежности сети.

Представители большинства опрошенных операторов затруднились назвать часть сети либо инфраструктуры, на которой можно ожидать первых внедрений. Ответившие операторы отметили, что первые внедрения ожидаются в следующих сегментах: пакетное ядро EPC, IMS, сегменты магистральной транспортной сети (прогноз касается внедрения NFV в сетевом домене), а также в ЦОДах операторов.

Как показывают опросы, примерно 2/3 российских специалистов отметили, что их интерес к SDN пока носит лишь чисто теоретический характер. Между тем, «использование принципов программируемого управления сетью и виртуализации сетевых сервисов для фор-

мирования проблемно-ориентированных вычислительных сред, предназначенных для решения сложных прикладных проблем» входит в правительственный «Перечень приоритетных научных задач, для решения которых требуется задействовать возможности федеральных центров коллективного пользования научным оборудованием». «Задача направлена на разработку комплекса сетевых и информационных технологий построения гибкой, адаптируемой под специфику исследований инфраструктуры на основе новой концепции организации сетевого пространства и вычислительных услуг» [13].

Из доклада J'son & Partners Consulting [14]:

К основным ожидаемым результатам относятся, в частности, «создание высокотехнологичной инфраструктуры для проведения исследований в области компьютерных сетей национального масштаба, развития Интернета нового поколения, апробации решений в области безопасности национального информационного пространства» и «построение распределенной платформы программируемого управления сетями, обеспечивающей отказоустойчивость и высокую доступность ресурсов, переорганизацию сетевых сервисов и сетевой инфраструктуры, согласованное планирование ресурсов».

В связи с тем, что на долю сетевого оборудования зарубежного производства в России приходится, по некоторым оценкам, более 90%, и в связи с ухудшением отношений с Западом задача замещения импорта стоит достаточно остро. В этом плане реализация успешных проектов в области SDN дает стране шансы стать весомым партнером лидеров данного сегмента ИТ-рынка.

В феврале 2012 г. на базе лаборатории вычислительных комплексов факультета ВМК МГУ был создан Центр прикладных исследований компьютерных сетей (ЦПИКС, резидент ИТ-кластера Фонда «Сколково»), в задачи которого входит проведение научных исследований в области сетевых технологий, в том числе SDN. В июле того же года ОАО «Ростелеком» заключило контракт с ЦПИ КС на проектирование и создание опытного сегмента облачной платформы для ЦОД на основе SDN. В мае 2014 г. «Ростелеком» начал работу над внедрением SDN и NFV.

Таким образом, концепции SDN и NFV в России находятся на стадии формирования. Для ускорения готовности к коммерческому внедрению необходимы:

- изучение и адаптация нормативно-правовых аспектов, технических требований и вопросов регулирования;
- создание ассоциаций научно-исследовательских университетов, лабораторий, профильных академических институтов, представителей телеком-сообщества, стартапов, российских

разработчиков;
- привлечение в Россию ведущих зарубежных экспертов в области SDN;
- интеграция российских исследователей и экспертов в международные проекты, связанные с SDN.

В целом, успех SDN в России и появление нового емкого сегмента рынка (программного обеспечения сетевых приложений) зависят от заинтересованности всех участников рынка ИКТ, в первую очередь, от отечественных ИТ-компаний и операторов, академических и отраслевых институтов, регулятора и других госструктур.

П.6. Список литературы

1. *Найденов Андрей*, эксперт по проектированию сетевой инфраструктуры компании IBM. Эволюция в сетях Дата-Центров. Программно-определяемые сети SDN // «Хабрахабр» — крупнейший в Европе ресурс для IT-специалистов:
   URL: http://habrahabr.ru/company/ibm/blog/211208
2. *Hao Yin, Yong Jiang, Chuang Lin, Yan Luo, Yunjie Liu* Big Data: Transforming the Design Philosophy of Future Internet // IEEE Network:
   URL: http://cloudcomputing.ieee.org (July/August 2014) P.14-19.
3. *Барсков А.* Cisco и SDN // «Журнал сетевых решений/LAN»:
   http://www.osp.ru/lan/2013/02/13033667
4. OpenFlow // Википедия — свободная энциклопедия:
   https://ru.wikipedia.org/wiki/Openflow
5. OpenFlow Switch Specification Version 1.5.0 (Protocol version 0x06) // Open Networking Foundation December 19, 2014.
6. *Sezer S., Scott-Hayward S., Chouhan P.K., et al.* Are We Ready for SDN? Implementation Challenges for Software-Defined Networks // IEEE Communications Magazine. — 2013. — Vol. 51, № 7. — P.36–43.
7. *Ефимушкин В.А., Языков Д.Н.* Анализ характеристик функционирования коммутатора программно-конфигурируемой сети: Доклад XII Всероссийского совещания по проблемам управления ВСПУ, 16–19 июня 2014 г. — Москва, 2014.
8. Network Functions Virtualization // Википедия — свободная энциклопедия: URL: http://en.wikipedia.org/wiki/Network_Functions_Virtualization (3 March 2015).
9. *Орлов Сергей* — ведущий редактор «Журнала сетевых решений/LAN». SDN и другие // Ethernet-форум: URL: http://www.osp.ru/iz/ethernet/articles/13041880 (2014.06.17)
10. *Kelly LeBlanc* What's The Difference Between SDN and NFV? //

VP of Marketing for 6WIND: URL: http://www.6wind.com/blog/whats-the-difference-between-sdn-and-nfv

11. *Prayson Pate* NFV and SDN: What's the Difference? // METASWITCH NETWORKS: URL: http://www.sdxcentral.com/articles/contributed/nfv-and-sdn-whats-the-difference/2013/03/ (2013, March 30).
12. *Ефимушкин В.А., Ледовских Т.В., Корабельников Д.М.* Международная стандартизация программно-конфигурируемых сетей // Электросвязь. — 2014. — № 8. — С.7.
13. Более половины операторов России — на этапе тестирования технологий SDN и NFV Российская служба IT российская служба IT новостей // Бестселлеры IT-рынка Аналитика российского рынка ИТ: URL: http://www.itbestsellers.ru/problems/detail.php?ID=30563 (30 сентября 2014 г.).
14. Новые тренды в мобильной связи: виртуализация сетевых функций (Network Functions Virtualization, NFV) и концепция программно-конфигурируемой сети (Software defined networking, SDN) // J'son & Partners Consulting — ведущая международная консалтинговая компания: http://json.tv/ict_telecom_analytics_view/novye-trendy-v-mobilnoy-svyazi-virtualizatsiya-setevyh-funktsiy-network-functions-virtualization-nfv-i-kontseptsiya-programmno-konfiguriruemoy-seti-software-defined-networking-sdn (23 Июля 2014 г.).

Термины и определения

АО — адрес отправителя.

АПД — аппаратура передачи данных.

АСР — асинхронно сбалансированный режим.

АП — адрес получателя.

ВОЛС — волоконно-оптическая линия связи.

ВОТК — выборочный отказ.

Временное разделение каналов (ВРК, TDM) — разделение каналов во времени, каждому каналу выделяется квант времени (тайм-слот).

ВС — вторичная станция.

ГП — готов к приему.

Децентрализованная сигнализация — способ обмена сигналами в процессе установления и разъединения соединений в сети с коммутацией каналов, при котором за каждым каналом для передачи речевой информации или данных закрепляется индивидуальный сигнальный канал.

Единичный интервал — минимальный интервал времени, которому равны значащие интервалы времени сигнала.

Единичный элемент — элемент сигнала, имеющий длительность, равную единичному интервалу времени.

Информация — сведения, являющиеся объектом передачи, распределения, преобразования, хранения или непосредственного использования.

ИОС — информационная обратная связь.

ИС — источник сообщений.

КАМ — Квадратурная (амплитудная) модуляция (Quadrature Amplitude Modulation, QAM) — разновидность амплитудной модуляции сигнала, которая представляет собой сумму двух несущих колебаний одной частоты, но сдвинутых по фазе относительно друг друга на 90°.

Канальный уровень (Data Link layer) — уровень сетевой модели OSI, предназначенный для передачи данных узлам, находящимся в том же сегменте локальной сети.

Код Хемминга — систематический код с кодовым расстоянием $d_0 = 3$ или $d_0 = 4$.

Кодовое расстояние — минимальное для данного кода расстояние Хемминга.

КК (коммутация каналов) — совокупность операций по соединению каналов для получения сквозного канала, связывающего через узлы коммутации один оконечный пункт с другим. При КК сначала организуется через узлы коммутации сквозной канал передачи сообщений между взаимодействующими абонентами, а затем осуществляется передача сообщений.

КО — кадр отвергнут.

Кодовое разделение каналов (КРК, CDMA) — разделение каналов по кодам, каждый канал имеет свой код, наложение которого на групповой сигнал позволяет выделить информацию конкретного канала.

Коллизия кадров — это наложение двух и более кадров (пакетов) от станций, пытающихся передать кадр в один и тот же момент времени из-за наличия задержки распространения сигнала по сети или наличия неисправной сетевой платы.

Коммутатор — это устройство, конструктивно выполненное в виде сетевого концентратора и действующего как высокоскоростной много портовый мост.

КП (коммутация пакетов) — разновидность коммутации с накоплением, при которой длинные сообщения передаются не целиком, а разбиваются на относительно короткие части — пакеты.

ЛВС — локально-вычислительные сети.

Маска подсети или маска сети — это битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети.

Многопротокольная поддержка — способность протокола PPP поддерживать несколько протоколов сетевого уровня.

НГП — не готов к приему.

НО — начальный ограничитель.

ООД — оконечное оборудование данных.

ОЗУ (RAM) — память с произвольным доступом. Энергозависимая память, то есть ее содержимое стирается после выключения питания маршрутизатора. Поэтому она используется для хранения промежуточных данных во время работы маршрутизатора.

ОС — обратная связь.

ОТК — отказ.

Пакет — часть сообщения, представленная в виде блока с заголовком, имеющего установленный формат (структуру данных) и ограниченную длину, и передаваемая по сети как единое целое.

ПД — передача данных.

ПЗУ (ROM) — постоянное запоминающее устройство, используется для хранения загрузочного программного обеспечения, которое запускается первым в момент включения маршрутизатора и в дальнейшем отвечает за его загрузку.

Помехоустойчивость линии — способность линии уменьшать уровень помех, создаваемых во внешней среде и на внутренних проводниках.

ППЗУ (NVRAM) — энергонезависимая память. Хранятся дополнительные конфигурации маршрутизатора, которые считываются при последующих загрузках.

Префикс — идентификатор оператора, сети, связи. Общая старшая часть адреса всех сетей каждого поставщика услуг.

Пропускная способность канала связи — максимально возможное значение скорости передачи информации по каналу связи.

Протоколы верхнего уровня (5–7) — протоколы семиуровневой ЭМВОС, ориентированные на обработку информации.

Протоколы нижнего уровня (1–4) — протоколы семиуровневой ЭМВОС, ориентированные на передачу информации.

ПС — первичная станция.

РАО — режим асинхронного ответа.

Расширяемость протокола — возможность включения новых протоколов в стек PPP, так и возможность применения собственных протоколов пользователей вместо рекомендуемых в PPP по умолчанию.

РЗД — разъединить.

РНО — режим нормального ответа.

РОС — решающая обратная связь.

РОС-ОЖ — решающая обратная связь с ожиданием.

Сетевой концентратор или **Хаб** (hub — центр деятельности) — сетевое устройство для объединения нескольких устройств Ethernet в общий сегмент.

Сеть транспортная — это совокупность ресурсов систем передачи (каналов, трактов, секций) и относящиеся к ним средства контроля, оперативного переключения, резервирования и управления, предназначенные для переноса информации между заданными пунктами сети.

Синхронизация — процесс установления и поддержания определенных временных соотношений между двумя или несколькими процессами.

Синфазность — обеспечение циклов фазирования, т.е. разделение кодовых комбинаций или символов первичного алфавита, а также отдельных частей кадра, пакета, фрагмента, блока сообщений.

Скорость передачи информации — число бит, передаваемых в секунду.

Скорость телеграфирования (скорость модуляции) — число единичных элементов, которое можно передать в секунду.

Скремблирование (от англ. scramble — переменчивость) — сложение цифрового сигнала по правилам двоичной арифметики с псевдослучайной двоичной последовательностью с целью исключения из сигнала длинных последовательностей нулей, изменения спектра сигнала и т.п.

Сообщение — форма представления информации.

Спектральное разделение каналов (СРК, WDM) — разделение каналов по длине волны.

СТ — системы телекоммуникаций.

УАСР — установить АСР.

УР — устройство разделения.

УРНО — установить режим нормального ответа.

Централизованная сигнализация — способ обмена сигналами в процессе установления и разъединения соединений в сети с коммутацией каналов, при котором все сигнальные сообщения для большой группы пользователей передаются в одном (общем) канале сигнализации.

Циклический код — разновидность систематического кода, основное свойство которого заключается в следующем. Если комбинация a_0, a_1, ..., a_{n-1} разрешенная, то комбинация, получаемая из нее путем циклической перестановки элементов, т.е. комбинация a_{n-1}, a_0, a_1, ..., a_{n-2}, также является разрешенной.

Цифровой сигнал — последовательность, состоящая из чередующихся случайным образом импульсов и пауз одинаковой длительности; при этом импульс обозначают 1, а паузу — 0.

ЦСП — цифровые системы передачи.

Частотное разделение каналов (ЧРК, FDM) — разделение каналов по частоте, каждому каналу выделяется определённый диапазон частот.

Шлюз — устройство, позволяющее организовать обмен данными между сетевыми объектами, использующими различные протоколы обмена данными. Шлюз выполняет свои функции на уровнях выше сетевого. Он не зависит от используемой

передающей среды, но зависит от используемых протоколов обмена данными. Как правило, шлюз выполняет преобразования между какими-либо двумя протоколами (например, NetWare и TCP/IP и т.д.).

Энтропия — мера неопределенности в поведении источника дискретных сообщений.

AAL (ATM Adaptation layer) — уровень адаптации ATM.

AMI (Bipolar Alternate Mark Inversion) — метод биполярного кодирования с альтернативной инверсией.

Anycast — идентификатор набора интерфейсов, принадлежащих разным узлам. Пакет, посланный по эникастному адресу, доставляется одному из интерфейсов, указанному в адресе (ближайшему маршрутизатору по пути следования пакета в соответствии с протоколом маршрутизации).

API (application programming interface) — интерфейс программирования приложений, интерфейс прикладного программирования.

ARP (Address Resolution Protocol, протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения MAC-адреса по известному IP-адресу.

AS (Autonomous systems, автономные системы) — выражение, используемое в протоколах маршрутизации по отношению к системам, находящимся в административном ведении и под единоличным контролем пользователя, группы или же организации. Каждая AS описывается своим номером ASN (AS Number). Например, номер ASN 1 зарегистрирован за компанией Genuity. Для перемещения трафика между AS обычно используются BGP, тогда как для маршрутизации данных в пределах отдельной AS — маршрутный протокол, например, OSPF.

ATM (Asynchronous Transfer Mode, асинхронный режим передачи) — высокоскоростная, ориентированная на соединение, коммутируемая и мультиплексирующая технология, которая использует ячейки размером 53 байта (5 байт — заголовок, 48 байт — полезная нагрузка) для одновременной передачи данных различных видов, включая голос, видео и собственно данных. Использование асинхронного режима означает, что информационные потоки могут быть посланы независимо от общего таймера. Технология ATM описывается архитектурой с тремя плоскостями:

– пользователя (Plane User, U) — координирует интерфейс между протоколами верхних уровней, такими как

IP и ATM;
- управления (Plane Management, M) — координирует все уровни стека протоколов ATM;
- сигнализации (Plane Control, C) — координирует процессы обмена сигнальными сообщениями, установления и разъединения виртуальных каналов и трактов;

ATM разработан для полного использования преимуществ высокоскоростных линий передачи, таких как SONET, E3 и T3.

AUI (Attachment Unit Interface) — интерфейс модуля присоединения.

BGP (Border Gateway Protocol, протокол граничного шлюза) — основной протокол динамической маршрутизации в Интернете. Протокол BGP предназначен для обмена информацией о достижимости подсетей между автономными системами.

Billing (биллинг) — тарификация, составление и выписывание счетов за предоставление телекоммуникационных услуг.

BPDU (Bridge Protocol Data Unit) — фрейм (единица данных) протокола управления сетевыми мостами, используется для исключения возможности возникновения петель в сетях передачи данных при наличии в них многосвязной топологии.

BSC (Binary Synchronous Communication) — протокол двоичной синхронной связи.

CHAP (Challenge Handshake Authentication Protocol) — протокол аутентификации по квитированию вызова.

CIDR (Classless Inter-Domain Routing) — бесклассовая адресация. Метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки классовой адресации. Использование этого метода позволяет экономно использовать ограниченный ресурс IP-адресов, поскольку возможно применение различных масок подсетей к различным подсетям.

CP (Control Protocol) — протокол управления.

CS (Convergence Sub-layer) — подуровень конвергенции ATM, отвечает за получение протокольного модуля данных (Protocol Data Unit, PDU) от вышележащих уровней и их адаптацию, обычно за счет добавления служебной информации для дальнейшего представления уровню SAR.

CSLIP (Compressed SLIP) — усовершенствованный SLIP, изменения коснулись сжатия IP-заголовков и TCP-заголовков. 40 байт этих двух заголовков могут сжиматься до 3–5 байт. CSLIP даёт заметный выигрыш против SLIP только при использовании небольших пакетов и хороших линий связи, так

как при необходимости повтора передачи в CSLIP заново переданы будут все пакеты, вплоть до последнего переданного несжатого, против одного пакета в SLIP.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) — множественный доступ с прослушиванием несущей и обнаружением столкновений.

CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance) — множественный доступ с контролем несущей и избеганием коллизий.

CWDM (coarse WDM, грубые WDM) — системы с частотным разносом каналов более 2500 ГГц, позволяющие мультиплексировать не более 18 каналов. Используемые в настоящее время CWDM работают в полосе от 1271 нм до 1611 нм, промежуток между каналами 20 нм (2500 ГГц), можно мультиплексировать 16 спектральных каналов.

DDCMP (Digital Data Communication Message Protocol) — байт-ориентированный протокол, разработан в фирме Digital Equipment Corporation (DEC). DDCMP предназначен для обеспечения синхронной работы по дуплексным и полудуплексным соединениям, устанавливаемым по коммутируемым или выделенным каналам, в сетях «от точки к точке» или многоточечным соединениям. Причем, в последнем случае одна станция является первичной (основной), а другие — вторичными (ведомыми).

Demand priority — метод доступа с запросом приоритета.

DiffServ (Differentiated Service) — дифференцированное обслуживание.

designated port — назначенный порт.

designated bridge — назначенный коммутатор.

DHCP (Dynamic Host Configuration Protocol, протокол динамической настройки узла) — сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

DNA (Digital Network Architecture) — цифровая сетевая архитектура.

DNS (Domain Name System, система доменных имён) — компьютерная распределённая система для получения информации о доменах.

DSL (Digital Subscriber Line) — цифровая абонентская линия. DSL является достаточно новой технологией, позволяющей значительно расширить полосу пропускания старых медных телефонных линий, соединяющих телефонные станции с индивидуальными абонентами. Объединяет несколько

технологий (ADSL, SDSL, SHDSL и т.д.).

DVLAN (Dynamic VLAN) — динамическая таблица VLAN.

DWDM (dense WDM, плотные WDM) — системы с разносом каналов около 100 ГГц, позволяющие мультиплексировать до 40 каналов.

ETB (End of Block) — конец блока (КБ).

ETC (Egress Tag Control) — Контроль признака на выходе.

Ethernet — стандарт, первоначально предложенный для локальных сетей, работающих со скоростью 10 Мбит/с; в нем для управления доступом к сети используется протокол CSMA/CD.

ETX (End of TeXt) — конец текста (КТ).

FCS (Frame Control Sequence) — контрольная сумма, используется для обнаружения ошибок передачи между двумя станциями.

FDDI (Fiber Distributed Data Interface) — технология передачи данных, использующая оптическое волокно. Стандарт американского института стандартов (ANSI), принятый без изменения ISO. Протокол рассчитан на физическую скорость передачи информации 100 Мбит/с и предназначен для сетей с суммарной длиной до 100 км (40 км для мультимодовых волокон) при расстоянии между узлами 2 км или более. Частота ошибок в сети не превышает 10^{-9}.

FEC (Forwarding Equivalence Class) — класс эквивалентности продвижения.

Frame Relay — ретрансляция кадров. Протокол, используемый для создания глобальных сетей, в котором данные между пунктами назначения передаются в виде кадров.

FTP (File Transfer Protocol, протокол передачи файлов) — стандартный протокол, предназначенный для передачи файлов по TCP-сетям (например, Интернет). FTP часто используется для загрузки сетевых страниц и других документов с частного устройства разработки на открытые сервера хостинга.

GFP (Generic Framing Procedure) — протокол общей процедуры формирования кадра.

GMPLS (Generalized multiprotocol label switching, обобщенная многопротокольная коммутация меток) — расширение функциональных возможностей MPLS с целью включения элементов, не охватываемых IP-протоколом (non-IP elements), таких как кросс-соединители, маршрутизаторы волновых каналов (wavelength routers) или оптические мультиплексоры добавления/ответвления каналов (add-drop multiplexers).

GVRP (GARP VLAN Registration Protocol) — сетевой протокол

канального уровня модели OSI/ISO, позволяющий устройству локальной сети сообщить соседним устройствам, что оно желает принять пакеты для одной или нескольких VLAN. Главная цель GVRP — позволить коммутаторам автоматически обнаружить информацию о VLAN, которая иначе должна была бы быть вручную сконфигурирована в каждом коммутаторе. Этого можно достичь использованием GVRP — распространить идентификаторы VLAN по локальной сети. GVRP также может быть использован сетевыми серверами. Эти серверы обычно конфигурируются для вхождения в несколько VLAN, и затем сообщают коммутаторам о VLAN, к которым они хотят присоединиться.

GUI (Graphical user Interface) — графический интерфейс пользователя.

HA (High Availability) — высокая готовность.

HDLC (High-Level Data Link Control) — протокол канального уровня, который обеспечивает передачу последовательности пакетов через физический канал, искажения в котором вызывают ошибки в передаваемых данных, потерю, дублирование пакетов и нарушения порядка прибытия пакетов к адресату.

HDSL (High Data Rate Digital Subscriber Line) — высокоскоростная цифровая абонентская линия.

HDWDM (high dense WDM, высокоплотные WDM) — системы с разносом каналов 50 ГГц и менее, позволяющие мультиплексировать более 64 каналов.

HOP (High Order Path Layer) — уровень тракта высокого порядка.

I — кадры (информационные), служат для переноса самой информации или данных.

ICMP (Internet Control Message Protocol, протокол межсетевых управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают.

IntServ (Integrated Service) — Интегрированный сервис.

IP — протокол сетевого уровня Internet.

IP address — IP-адрес. Часто называется Интернет-адресом. Адрес, однозначно определяющий любое устройство (хост) в Интернете (или любой сети TCP/IP). Каждый адрес состоит из четырех октетов (32 бита), записанных в виде десятичных

чисел и разделенных запятыми. Адрес состоит из номера сети, необязательного номера подсети и номера хоста. Номера сети и подсети применяются для маршрутизации, а номер хоста определяет хост в сети или подсети. Информация о сети и подсети выделяется из IP-адреса с помощью маски подсети. IP-адреса делятся на пять классов (A–E), в каждом из которых на сеть, подсеть и хост отводится строго определенное число разрядов. См. также: CIDR, IP и subnet mask.

IP datagram (дейтаграмма протокола IP) — основная единица информации.

IS-IS — протокол маршрутизации промежуточных систем. Это протокол внутренних шлюзов (IGP), стандартизированный ISO и использующийся в основном в крупных сетях провайдеров услуг. IS-IS может также использоваться в корпоративных сетях особо крупного масштаба. IS-IS — это протокол маршрутизации на основе состояния соединений. Он обеспечивает быструю сходимость и отличную масштабируемость. Как и все протоколы на основе состояния соединений, IS-IS очень экономно использует пропускную способность сетей.

ISDN (Integrated Services Digital Network) — цифровая сеть с интеграцией служб.

ISO (International Organization for Standardization) — Международная организация по стандартизации.

ISN (Initial Sequence Number) — начальный номер последовательности.

LAN — технологии локальных сетей.

LAPB (Link Access Procedure Balanced) — сбалансированная процедура доступа к звену передачи данных (применяется в стандарте X.25).

LAPD (Link Access Procedure D-channel) — предназначен для управления звеном в цифровых сетях с интеграцией служб ISDN (Integrated Services Digital Network) — система, в которой по телефонным каналам передаются только цифровые сигналы, в том числе и по абонентским линиям, то есть конечный абонент передает данные непосредственно в цифровой форме.

LAPM (Link Access Protocol for Modems) — протокол коррекции ошибок в стандарте V.42 для модемов и предназначен для коммутируемой телефонной сети PSTN (Public Switched Telephone Network).

LAPX (расширенный LAPB) — используется в терминальных системах и в стандарте телетекса. Является полудуплексным

вариантом HDLC.

LCP (Link Control Protocol) — протокол управления линией связи.

LDAP (Lightweight Directory Access Protocol, облегчённый протокол доступа к каталогам) — относительно простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации, поиска и сравнения, а также операции добавления, изменения или удаления записей.

LDP (Label Distribution Protocol) — протокол обмена метками.

LER (Label edge router, краевой маршрутизатор меток) — маршрутизатор, инициирующий LSP в сети MPLS; пограничный маршрутизатор домена MPLS.

LLC (Link layer control) — уровень управления логической передачей данных.

Loopback — адрес замыкания.

LSA (link-state advertisement) — объявление о состоянии канала, описывает все каналы маршрутизатора, все интерфейсы и состояние каналов.

LSP (Label switched path, коммутируемый посредством меток маршрут) — обеспечиваемый между двумя маршрутизаторами поток пакетов MPLS. В общих чертах LSP аналогичны каналам в технологиях ATM и Frame Relay; путь коммутации пакетов с помощью меток.

LSR (Label switched router, маршрутизатор с коммутацией меток) — один из маршрутизаторов MPLS, устанавливаемых между LER, обеспечивающий создание LSP; транзитный маршрутизатор домена MPLS, коммутирующий пакеты с помощью меток.

MAC (Media Access Control, управление доступом к среде) — протокол, используемый для определения способа получения доступа рабочих станций к среде передачи, наиболее часто используемый в локальных сетях. Для ЛВС, соответствующих стандартам IEEE, MAC-уровень является нижним подуровнем канала передачи данных (data link layer); управление доступом к среде передачи данных (нижний подуровень уровня звена данных эталонной модели OSI).

MAC address (MAC-адрес) — аппаратный адрес канального уровня, необходимый каждому порту или устройству для соединения с участком LAN. Эти адреса используются разными устройствами сети для точного определения места логических адресов. MAC-адреса определяются стандартом IEEE. Они состоят из шести символов, как правило, используя впаянный

(BIA-) адрес локального интерфейса LAN. Имеет много названий: «аппаратный адрес», «физический адрес», «впаянный адрес» или «адрес уровня MAC» (подуровня управления доступом к среде).

MAN (Metropolitan Area Network) — региональная или городская вычислительная сеть. Любая сеть, объединяющая область, приблизительно равную большому городу. Как правило, такие сети крупнее LAN и меньше WAN. См. также: LAN.

MII (Media Independet Interface) — интерфейс независящий от среды передачи.

MCP (multi-path constrained) — маршрутизация пути с множеством ограничений.

MDI (Medium Dependent Interface) — интерфейс, зависящий от среды передачи.

MLPPP (Multi Link PPP) — многоканальный протокол PPP.

MPLS (Multiprotocol label switching, многопротокольная коммутация меток) — определяет следующее поколение протокола маршрутизации, в котором решения о передаче данных в сети принимаются на основе анализа коротких меток, внедренных в пакеты, а не длинных сетевых адресов; многопротокольная коммутация с помощью меток.

MS (Multiplex Section) — секция мультиплексирования.

MTU (maximum transmission unit) — максимальный размер полезного блока данных одного пакета, который может быть передан протоколом без фрагментации.

Multicast — Идентификатор набора интерфейсов, принадлежащих разным узлам. Пакет, посланный по мультикастинг-адресу, доставляется всем интерфейсам, заданным этим адресом.

NAT (Network Address Translation) — функция NAT заменяет частные IP-адреса открытыми зарегистрированными IP-адресами в каждом пакете протокола IP.

NCP (Network Control Protocol) — протокол управления сетью.

NGN (Next Generation Network) — сеть связи следующего поколения.

NGSDH (Next Generation SDH) — система SDH нового поколения.

NMS (Network Management System) — система, в составе которой используется оборудование и программы, используемые для мониторинга, управления и администрирования сети передачи данных.

NRZ (Non Return to Zero, без возврата к нулю) — это простейший код, представляющий собой обычный цифровой сигнал.

NRZI (Non Return to Zero Inverted) — инверсное кодирование без возврата к нулю.

OCh (Optical Channel) — оптический канал.

ODUk (Optical Data Unit k) — блоки данных оптических каналов.

OFDM (Orthogonal frequency-division multiplexing) — мультиплексирование с ортогональным частотным разделением каналов, является цифровой схемой модуляции, которая использует большое количество близко расположенных ортогональных поднесущих.

OMS (Optical Multiplex Section) — оптическая секция мультиплексирования.

OMX (Optical Multiplex) — оптический мультиплексор.

OPUk (Optical Channel Payload Unit k) — блоки полезной нагрузки оптических каналов.

OSI (Open System Interconnection) — взаимодействие открытых систем (ВОС).

OSPF (Open Shortest Path First) — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала (link-state technology) и использующий для нахождения кратчайшего пути алгоритм Дейкстры.

OTM (Optical Transport Module) — оптические волновые (транспортные) модули.

OTN (Optical Transport Network) — оптические транспортные сети.

OTS (Optical Transmission Section) — оптические секции ретрансляции.

OTUk (Optical Data Unit k) — цифровые оптические транспортные блоки.

Path Cost — стоимость пути.

PAP (Password Authentication Protocol) — протокол аутентификации по паролю.

PBT (Provider Backbone Transport) — операторский транспорт в опорной сети.

PCS (Physical Coding Sublayer) — верхний подуровень физического кодирования.

PDH (Plesiochronous Digital Hierarchy, плезиохронная цифровая иерархия) — европейский стандарт для волоконно-оптических сетей.

PE (Provider Edge Router) — граничный маршрутизатор провайдера.

PHB (Per Hop Behaviour) — независимое поведение маршрутизаторов.

PHY (Physical layer) — физический уровень.

PMA (Physical Medium Attachment) — подуровень подсоединения к физической среде.

PMD (Physical Medium Dependent) — подуровень физического уровня, зависящий от среды передачи.

PPP (Point-to-Point Protocol) — обеспечивает связь с удаленными сетями через стандартный PPP-сервер. Протокол PPP также позволяет серверу удаленного доступа принимать входящие вызовы от программ удаленного доступа других разработчиков, поддерживающих PPP.

Protocols (протоколы) — набор правил, определяющих способ передачи информации между устройствами.

PSTN (Public Switched Telephone Network) — коммутируемая телефонная сеть общего пользования.

PVID (Port VLAN Indentifier) — идентификатор, определяющий принадлежность порта к конкретной виртуальной сети внутри коммутатора.

QoS (Quality of Service, качество обслуживания) — набор параметров для измерения качества передачи и доступности службы какой-либо системы передачи.

RARP (Reverse Address Resolution Protocol, обратный протокол преобразования адресов) — протокол сетевого уровня модели OSI, выполняет обратное отображение адресов, то есть преобразует физический адрес в IP-адрес.

RIP (Routing Information Protocol) — является дистанционно-векторным протоколом внутренней маршрутизации. Применяется в небольших компьютерных сетях, позволяет маршрутизаторам динамически обновлять маршрутную информацию (направление и дальность в хопах), получая ее от соседних маршрутизаторов.

ROADM (Reconfigurable Optical Add-Drop Multiplexer) — технология переконфигурируемого оптического мультиплексирования ввода/вывода.

root port — корневой порт.

Routing (маршрутизация) — процесс пересылки пакетов с логическими адресами из их локальной подсети к конечному пункту назначения. В крупных сетях многочисленные промежуточные назначения. Иногда пакет проходит их до того, как дойдет до своего назначения.

Routing Protocol (протокол маршрутизации) — протокол, определяющий алгоритмы обновления таблиц маршрутизации между маршрутизаторами. Примерами таких протоколов могут служить RIP, IGRP и OSPF.

Routing Protocols (протоколы маршрутизации) — сетевое программное обеспечение, применяемое маршрутизаторами для передачи коллективно используемой информации по всей топологии маршрутов с целью выбора оптимального маршрута для перемещения пакетов по сети.

Routing table — таблица маршрутизации или таблица маршрутов. Таблица, хранящаяся в маршрутизаторе или другом устройстве поддержки сетевого комплекса, которое поддерживает запись только лучших из возможных маршрутов до определенных сетевых назначений и связанные с этими маршрутами метрики.

RPR (Resilient Packet Ring) — протокол защищаемого пакетного кольца или пакетного кольца с самовосстановлением.

RS (Regenerator Section) — регенерационная секция.

RS (Reconciliation Sublayer, подуровень согласования) — функция отображения, согласующая сигналы перед интерфейсом Gigabit Media Independent Interface (XGMII), соединяющим с подуровнем MAC и PCS.

RS-232 (Recommended Standard 232) — физический уровень асинхронного (UART) интерфейса. Исторически имел широкое распространение в телекоммуникационном оборудовании для персональных компьютеров. В настоящее время всё ещё широко используется для подключения всевозможного специального или устаревшего оборудования к компьютерам, однако в основном он уже вытеснен интерфейсом USB.

RSTP (Rapid STP) — быстрый протокол покрывающего дерева.

RSVP (Resource ReSerVation Protocol) — протокол резервирования сетевых ресурсов.

RTCP (Real-Time Transport Control Protocol, протокол управления передачей в реальном времени) — протокол, используемый совместно с RTP. RTCP базируется на периодической передаче управляющих пакетов всем участникам сессии, используя тот же механизм рассылки, что и для пакетов данных. Протокол RTCP используется для передачи информации о задержках и потерях медиа-пакетов, джиттер-буфере, уровне звукового сигнала.

RTP (Real-time Transport Protocol) — используется при передаче трафика реального времени, переносит в своём заголовке данные, необходимые для восстановления аудиоданных или видеоизображения в приёмном узле, а также данные о типе кодирования информации.

PVC (Permanent Virtual Connection) — постоянное виртуальное соединение, долговременное соединение (несколько дней и даже месяцев), которое обычно устанавливается между конечным оборудованием сети ATM и используется при работе операторов.

S — кадры (супервизорные), используются для восстановления кадров, потерянных из-за искажений в канале, а также для управления потоками данных.

SAR (Segmentation And Reassembly) — подуровень сегментации и восстановления ATM. Задачей подуровня SAR является формирование модулей длиной 48 октетов, которые становятся полезной нагрузкой ячеек ATM.

SDH (Synchronous Data Hierarchy) — европейский стандарт цифровой системы передачи, ориентированный на использование оптических кабелей в качестве физической среды передачи данных для высокоскоростных сетей передачи на значительные расстояния; европейский эквивалент SONET.

SDLC (Synchronous Data Link Control) — синхронное управление звеном данных, разработан компанией IBM для системной сетевой архитектуры SNA.

SDN (Soft Defined Networks, программно-конфигурируемые сети, ПКС) — сеть передачи данных, в которой уровень управления сетью отделён от устройств передачи данных и реализуется программно, одна из форм виртуализации вычислительных ресурсов. Ключевые принципы программно-конфигурируемых сетей: разделение процессов передачи и управления данными, централизация управления сетью при помощи унифицированных программных средств, виртуализация физических сетевых ресурсов. Протокол OpenFlow, реализующий независимый от производителя интерфейс между логическим контроллером сети и сетевым транспортом, является одной из реализаций концепции программно-конфигурируемой сети и считается движущей силой её распространения и популяризации.

SLA (Service Level Agreement) — соглашение об уровне предоставления услуги, обозначающее формальный договор между заказчиком услуги и её поставщиком, содержащее описание услуги, права и обязанности сторон и, самое главное, согласованный уровень качества предоставления данной услуги.

SLIP (Serial Line Internet Protocol) — устаревший сетевой протокол канального уровня эталонной сетевой модели OSI для доступа к сетям стека TCP/IP через низкоскоростные линии связи путём простой инкапсуляции IP-пакетов. Используются

коммутируемые соединения через последовательные порты для соединений клиент–сервер типа «точка–точка».

SNA (Systems Network Architecture) — системная сетевая архитектура.

SOH (Start of Heading) — начало заголовка (НЗ).

SONET (Synchronous optical network, синхронная оптическая сеть) — широко распространенный стандарт для волоконно-оптических линий связи.

STA (Spanning-Tree Algorithm) — алгоритм покрывающего дерева, позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой.

STP (Shielded Twisted Pair) — экранированная витая пара.

STP (Spanning Tree Protocol) — протокол связующего дерева, канальный протокол. Основной задачей STP является устранение петель в топологии произвольной сети Ethernet, в которой есть один или более сетевых мостов, связанных избыточными соединениями. STP решает эту задачу, автоматически блокируя соединения, которые в данный момент для полной связности коммутаторов являются избыточными.

STX (Start of TeXt) — начало текста (НТ).

SVLAN (Static VLAN) — статическая таблица VLAN.

SYN (Synchronous Idle) — синхросимвол.

T-MPLS (Transport Multiprotocol Label Switching, транспортная многопротокольная коммутация по меткам) представляет собой технологию, разработанную специально для применения в пакетных транспортных сетях операторов связи.

Telnet — сетевой теледоступ; протокол виртуального терминала в наборе протоколов Internet, который позволяет пользователям одного хоста подключаться к другому удаленному хосту и работать с ним как через обычный терминал.

TCP (Transmission Control Protocol) — протокол управления передачей транспортного уровня в стеке протоколов TCP/IP.

TCP/IP (Transmission control protocol / Internet protocol, протокол управления передачей / Internet протокол) — многоуровневая архитектура, являющаяся предшественником модели OSI и в настоящее время лежащая в основе Internet и большей части корпоративного трафика.

Token Ring — технология локальной вычислительной сети (LAN) кольца с «маркерным доступом».

TSAP (Transport Service Access Point) — адрес на транспортном уровне.

TTL (Time to life) — время жизни пакета в сети.

U — кадры (ненумерованные), используются для установления соединения и разъединения, завершения соответствующих режимов канала передачи.

ULA (unique local address) — уникальный локальный адрес устройства.

UNI (User network interface, сетевой интерфейс пользователя) — программное обеспечение, скрывающее сложность провайдерской сети при взаимодействии с нею снаружи. Клиенты могут использовать UNI для формирования запросов на сетевое обслуживание, но не могут проникнуть в нее для ознакомления с частной информацией.

Unicast — идентификатор одиночного интерфейса. Пакет, посланный по уникастному адресу, доставляется интерфейсу, указанному в адресе.

UDP (User Datagram Protocol) — протокол пользовательских дейтаграмм транспортного уровня в стеке протоколов TCP/IP.

UTP (Unshielded Twisted Pair) — неэкранированная витая пара.

VC (Virtual Container) — виртуальный контейнер, в них помещаются блоки данных, а также служебная информация для транспортировки.

VCI (Virtual Channel Identifier) — идентификатор виртуального канала.

VLAN (Virtual local area Network) — виртуальные локальные сети.

VLSM (Variable Length Subnet Masking) — переменная длина маски подсети.

VPC (Virtual Path Connection) — соединение виртуального пути.

VPI (Virtual path identifier, идентификатор виртуального пути) — поле ячейки ATM, определяющее маршрут, которому принадлежит ячейка; идентификатор виртуального тракта.

WAN (Wide Area Network) — технологии глобальных сетей.

Wi-Fi (Wireless Fidelity) — стандарт беспроводной передачи данных по радиоканалу уровня LAN.

WiMax (Worldwide Interoperability for Microwave Access) — стандарт беспроводного широкополосного доступа уровня города (Metropolitan).

WIS (WAN Interface Sublayer) — подуровень интерфейса глобальной сети.

X.25 — рекомендации ITU-T, определяющие стандарты для коммуникационных протоколов доступа к сетям с коммутацией пакетов (Packet Data Networks, PDN).

XGMII (10 Gigabit Media Independent Interface) — 10-гигабитный интерфейс, независимый от среды.

1000Base-T (Gigabit Ethernet) — спецификация для сетей Ethernet со скоростью передачи до 1000 Мбит/с на основе неэкранированной витой пары (unshielded twisted pair UTP) и волоконно-оптического кабеля.

100Base-T (Fast Ethernet) — спецификация для сетей Ethernet со скоростью передачи до 100 Мбит/с на основе неэкранированной витой пары (unshielded twisted pair, UTP) и волоконно-оптического кабеля.

10Base-T — спецификация для сетей Ethernet со скоростью передачи до 10 Мбит/с на основе неэкранированной витой пары (unshielded twisted pair, UTP) и волоконно-оптического кабеля.

100VG-AnyLAN — технология для любой сети, создана путем объединения стандарта Ethernet и Token Ring.

Оглавление