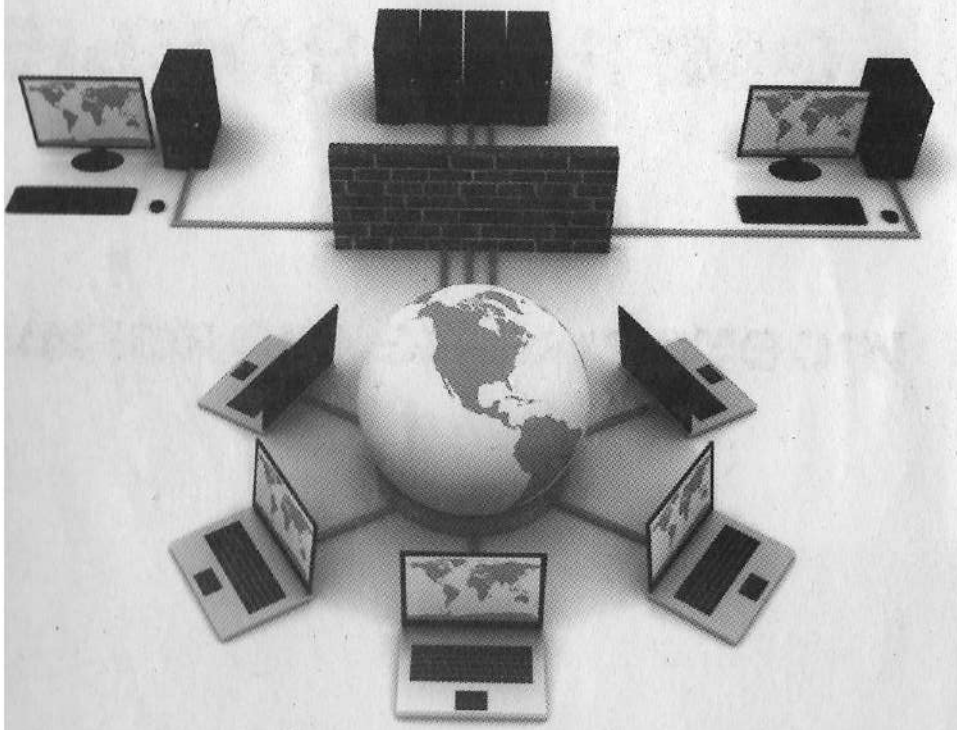


Кузьменко Н. Г.

Компьютерные сети

и сетевые технологии

НиТ
издательство
Наука и Техника
Санкт-Петербург

Кузьменко Н. Г.

Компьютерные сети и сетевые технологии. — СПб.: Наука и Техника, 2013. — 368 с.: ил.

Серия «Просто о сложном»

Данная книга представляет собой некоторое краткое энциклопедическое (учебное) издание, охватывающее очень широкий круг вопросов, связанных с организацией сетей передачи данных, начиная с азов – понятий, что такое компьютерные сети, состав сетей, протоколы и т.д. Автор постепенно переходит от одного понятия к другому и по ходу дела подробно останавливается на вопросах напрямую не связанных с архитектурой сетей.

По ходу изложения поясняются все необходимые понятия и технологии: принципы сжатия информации, назначение многих протоколов, технологии виртуальных сетей и принципы их реализации в коммутаторах, а также многое другое. Отдельное внимание уделено маршрутизаторам, без понимания методов построения таблиц маршрутной информации и их назначения нет смысла останавливаться на протоколах их реализации.

Материал книги основан на многолетнем опыте чтения автором лекций по нескольким смежным курсам, охватывающим весь спектр предлагаемого вашему вниманию материала.

Контактные телефоны издательства:
(812) 412 70 25, (812) 412 70 26, (044) 516 38 66
Официальный сайт: www.nit.com.ru

© Кузьменко Н. Г., 2013

Содержание

ВВЕДЕНИЕ 13
Для кого эта книга 13
Структура книги 14

ГЛАВА 1. ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ СЕТИ 17
1.1. ГЛОБАЛЬНЫЕ СЕТИ 18
1.2. ЛОКАЛЬНЫЕ СЕТИ 19
1.3. КАМПУСНЫЕ СЕТИ 22
1.4. ОБМЕН ИНФОРМАЦИЕЙ ЭВМ С ПЕРИФЕРИЙНЫМ УСТРОЙСТВОМ 23
1.5. ВЗАИМОДЕЙСТВИЕ ДВУХ КОМПЬЮТЕРОВ 27
1.6. МОДЕЛЬ OSI 31
1.7. СТАНДАРТНЫЕ СТЕКИ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ 36
Стек OSI 36
Стек TCP/IP 37
Стек IPX/SPX 38
Стек NetBIOS/SMB 39
Стек DECnet 39
Контрольные вопросы 40

ГЛАВА 2. КОМПОНЕНТЫ СЕТИ 42
2.1. РАБОЧИЕ СТАНЦИИ 43
2.2. СЕРВЕРЫ 44
Достаточная производительность 44
Большое дисковое пространство 45
Достаточный объём ОЗУ 45
Высокая надёжность 46
КОНТРОЛЬНЫЕ ВОПРОСЫ 63

ГЛАВА 3. КОММУНИКАЦИОННОЕ ОБОРУДОВАНИЕ 65
3.1. ЛИНИИ СВЯЗИ 66
3.2. ХАРАКТЕРИСТИКИ ЛИНИЙ СВЯЗИ 67

Коаксиальный кабель ... 72
Витая пара ... 74
Волоконно-оптический кабель ... 75

3.4. БЕСПРОВОДНЫЕ ЛИНИИ СВЯЗИ ... 79
Импульсная модуляция ... 81
Диапазоны частот ... 83

3.5. ОБЪЕДИНЕНИЕ КОМПЬЮТЕРОВ В СЕТЬ ... 85

КОНТРОЛЬНЫЕ ВОПРОСЫ ... 88

А 4. МЕТОДЫ ПЕРЕДАЧИ ДАННЫХ ... 90

4.1. КОДЫ, ПРИМЕНЯЕМЫЕ В СЕТЯХ ... 92
Потенциальный код ... 92
Биполярный код с инверсией единицы ... 93
Биполярный импульсный код ... 94
Манчестерский код ... 94
Потенциальный код 2B1Q ... 95

4.2. ЛОГИЧЕСКОЕ КОДИРОВАНИЕ ... 95
Избыточные коды ... 95
Скремблирование ... 97

4.3. ОБНАРУЖЕНИЕ И КОРРЕКЦИЯ ОШИБОК ... 99

4.4. СЖАТИЕ ИНФОРМАЦИИ ... 101
Текстовая информация ... 102
Графическая информация ... 103
Звуковая информация ... 106

КОНТРОЛЬНЫЕ ВОПРОСЫ ... 112

А 5. АРХИТЕКТУРА ETHERNET ... 114
Возникновение конфликтов ... 119
Время двойного оборота и распознавание конфликтов ... 121
Максимальная производительность сети Ethernet ... 124

5.1. ТОЛСТЫЙ ETHERNET ... 125

5.2. ТОНКИЙ ETHERNET ... 128

5.4. БЫСТРЫЙ (FAST) ETHERNET ... 133

5.5. СКОРОСТНАЯ ТЕХНОЛОГИЯ GIGABIT ETHERNET ... 135

5.6. ВЫСОКОСКОРОСТНАЯ ТЕХНОЛОГИЯ 10GIGABIT ETHERNET ... 137

5.7. ПЕРСПЕКТИВЫ РАЗВИТИЯ ТЕХНОЛОГИИ ETHERNET ... 139

5.8. ТЕХНОЛОГИЯ 40GIGABIT ETHERNET ... 141

5.9. ТЕХНОЛОГИЯ 100GIGABIT ETHERNET ... 142

КОНТРОЛЬНЫЕ ВОПРОСЫ ... 143

ГЛАВА 6. КОММУТАТОРЫ ... 145

6.1. МОСТЫ ... 147

6.2. КОММУТИРУЮЩАЯ СИСТЕМА ... 152
Общая шина ... 152
Коммутационная матрица ... 153
Многовходовая разделяемая память ... 154

6.3. КОНСТРУКТИВНОЕ ВЫПОЛНЕНИЕ КОММУТАТОРОВ ... 156
Законченное устройство ... 157
Стековый коммутатор ... 159
Модульный коммутатор ... 160

6.4. ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИИ КОММУТАТОРОВ ... 162
6.4.1. Идентификация источника ... 163
6.4.2. Подавление источника ... 164

КОНТРОЛЬНЫЕ ВОПРОСЫ ... 167

ГЛАВА 7. СВЯЗУЮЩЕЕ ДЕРЕВО ... 169

7.1. РАБОТА ПРОТОКОЛА СВЯЗУЮЩЕГО ДЕРЕВА STP ... 172

7.2. ТАЙМЕРЫ BPDU ... 176

7.3. СТОИМОСТЬ КОРНЕВОГО ПУТИ ... 178

7.4. АЛГОРИТМ РАБОТЫ ПРОТОКОЛА STP ... 179
Определение корневого коммутатора ... 179
Определение корневых портов ... 180

Перерасчет связей связующего дерева 185

7.5. ПРОТОКОЛ RSTP 186

7.6. ПРОТОКОЛ АГРЕГИРОВАНИЯ ПОРТА 189

КОНТРОЛЬНЫЕ ВОПРОСЫ 191

ВА 8. ВИРТУАЛЬНЫЕ ЛОКАЛЬНЫЕ СЕТИ 193

8.1. СТАТИЧЕСКИЕ ВИРТУАЛЬНЫЕ СЕТИ 197

8.2. ДИНАМИЧЕСКИЕ ВИРТУАЛЬНЫЕ СЕТИ 198

8.3. ВИРТУАЛЬНЫЕ СЕТИ, ОСНОВАННЫЕ НА ПРОТОКОЛАХ 199

8.4. КОНФИГУРИРОВАНИЕ ВИРТУАЛЬНЫХ СЕТЕЙ 203

8.5. МАГИСТРАЛЬНЫЙ ПРОТОКОЛ VTP 204

8.5.1. Объявления протокола VTP 208

8.5.2. Режимы коммутаторов в протоколе VTP 211

8.5.3. Отключение каналов в протоколе VTP 211

8.5.4. Настройка протокола VTP 212

8.6. ПРОТОКОЛ MSTP 214

8.7. ИЕРАРХИЧЕСКАЯ МОДЕЛЬ СЕТИ 217

8.7.1. Коммутаторы уровня доступа 219

8.7.2. Коммутаторы уровня распределения 220

8.7.3. Коммутаторы магистрального уровня 221

КОНТРОЛЬНЫЕ ВОПРОСЫ 224

А 9. ЧАСТНЫЕ ВИРТУАЛЬНЫЕ СЕТИ 226

9.1. ТИПЫ ЧАСТНЫХ ВИРТУАЛЬНЫХ СЕТЕЙ 227

9.2. РЕАЛИЗАЦИЯ ЧАСТНЫХ ВИРТУАЛЬНЫХ СЕТЕЙ 231

9.3. ПРОЕКТИРОВАНИЕ VPN-СЕТЕЙ 232

9.4. ПРОЦЕСС ВИРТУАЛЬНОГО УДАЛЁННОГО ДОСТУПА 235

Особые свойства службы виртуального удалённого доступа 237

ГЛАВА 10. СИСТЕМА АДРЕСАЦИИ В IP-СЕТЯХ 241

10.1. IP-АДРЕС 242

10.2. СТРУКТУРА IP-ПАКЕТА 248

10.3. МАСКИ ПОСТОЯННОЙ ДЛИНЫ 251

10.4. МАСКИ ПЕРЕМЕННОЙ ДЛИНЫ 255

10.5. ОБЕСПЕЧЕНИЕ НАДЁЖНОСТИ ПЕРЕДАЧИ ДАННЫХ 259

10.6. СТРУКТУРА IP-ПАКЕТА IPV6 262

КОНТРОЛЬНЫЕ ВОПРОСЫ 267

ГЛАВА 11. МАРШРУТИЗАТОРЫ 269

11.1. ДИСТАНЦИОННО-ВЕКТОРНЫЙ АЛГОРИТМ БЕЛЛМАНА-ФОРДА 278

Метод счёта до бесконечности 285

Метод расщепления горизонта 285

Метод обратного обновления 286

Метод таймеров удержания 286

Метод мгновенных обновлений 287

11.2. АЛГОРИТМ КРАТЧАЙШЕГО ПУТИ 289

11.3. ПРОТОКОЛ OSPF 293

11.4. ТИПЫ МАРШРУТИЗАТОРОВ В ПРОТОКОЛЕ OSPF 296

11.5. КОММУТАТОРЫ 3-ГО УРОВНЯ 298

Коммутация между виртуальными сетями на внешнем маршрутизаторе 299

Коммутатор с маршрутизацией виртуальных сетей 302

КОНТРОЛЬНЫЕ ВОПРОСЫ 308

ГЛАВА 12. ПРОТОКОЛЫ TCP И UDP 310

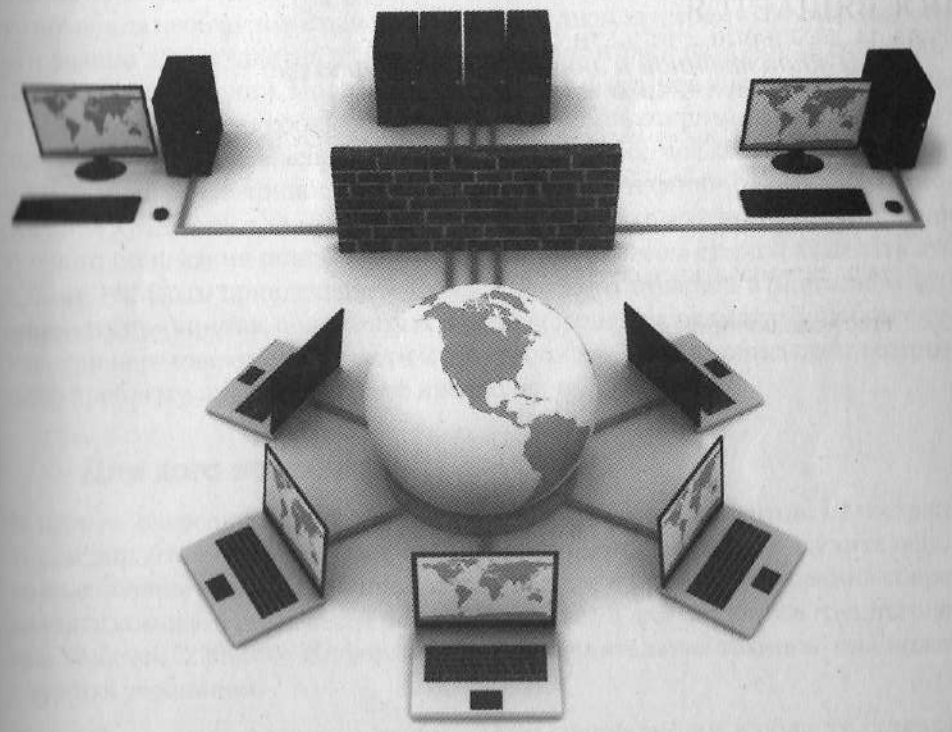
12.1. УСТАНОВКА И РАЗРЫВ СОЕДИНЕНИЯ 319

12.2. ВЫБОР ПЕРВОНАЧАЛЬНОГО НОМЕРА ОЧЕРЕДИ 320

12.3. ТАЙМЕРЫ ПРОТОКОЛА TCP 321

Таймер повторной передачи 322

Таймер время жизни сегмента 324
12.4. РАЗМЕР ОКНА 325
12.5. ПРОТОКОЛ UDP 328
КОНТРОЛЬНЫЕ ВОПРОСЫ 330
ГЛАВА 13. ОТКАЗОУСТОЙЧИВЫЕ СЕТИ 332
КОНТРОЛЬНЫЕ ВОПРОСЫ 342
ГЛАВА 14. БЕСПРОВОДНЫЕ СЕТИ 344
14.1. БЕСПРОВОДНЫЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ 347
14.1.1. Персональные беспроводные сети – Bluetooth 347
14.1.2. Локальные беспроводные сети – Wi-Fi 350
14.1.3. Беспроводные сети города – WiMAX 354
14.2. СЕТИ МОБИЛЬНОЙ СВЯЗИ 356
КОНТРОЛЬНЫЕ ВОПРОСЫ 365



ПОСВЯЩАЕТСЯ

SPecioIL
fOR
CODEьУ

PROXY
NiNJA

Введение

Эта книга является результатом многолетнего опыта преподавания автором курсов связанных с сетями на кафедре Вычислительной техники Красноярского политехнического института, впоследствии Института космических и информационных технологий Сибирского федерального университета. Автор стоял у истоков внедрения курса "Информатика" в средние школы СССР (1985 – 1986 годы), с этого момента и началось широкомасштабная подготовка в ВУЗах специалистов по сетям. Поскольку знание информационных технологий и умение их принять в жизни вышли за рамки узких специалистов и стали культурной составляющей специалиста в любой области, то возникла необходимость в простой, доступной но строгой форме донести эти знания для студентов разного уровня подготовки различных специальностей и направлений. Поэтому с учётом этих факторов и написана эта книга. Она представляет собой базовый курс по компьютерным сетям и охватывает практически все аспекты создания последних, созданный по принципу разумной доста-точности. То есть автор при описании отдельных технологий углубляется в технические детали на столько, сколько требуется для полного понимания описываемого предмета, при этом старался сделать это с мини-мальным привлечением математического аппарата и описанием физических принципов, положенных в основу данного предмета. В любом случае, при переходе от одной части к другой, автор сначала описывает возникшую проблему, а затем способ её преодоления.

Для кого эта книга

В первую очередь эта книга предназначена для всех студентов, магистрантов, аспирантов и технических специалистов которые хотят получить основополагающие знания о принципах построения и функционирования современных компьютерных сетей передачи данных, а для студентов группы специальностей "220000 – Информатика и вычислительная техника" она может служить учебником.

Книга будет так же полезна начинающим специалистам в области сетевых технологий, которые имеют только общие, весьма смутные представления, о работе сетей и хотят их расширить.

Особую помощь данная книга может оказать руководителям всех рангов, отдалённо знакомых с компьютерными сетями, в качестве источника овладения сетевой грамотностью.

Книгу, как и любой другой печатный документ, можно читать в любой по-

Структура книги

Книга состоит из введения 14 глав.

В **первой главе** рассматриваются самые общие вопросы построения вычислительных сетей. Даются определения сетей разного уровня сложности. На примере передачи информации между компьютером и печатающим устройством показан весь комплекс проблем, который возникает при организации передачи информации. А на примере передачи информации из приложения одного компьютера на устройство другого компьютера, показана сложность реализации взаимодействия между программными и аппаратными средствами. Для стандартизации взаимодействия между программными и аппаратными средствами разных уровней была разработана модель взаимодействия открытых систем – OSI, которая послужила основой для разработки сетевых устройств и приложений.

Вторая глава посвящена описанию основных компонентов сети. Осо-бое внимание уделено машинам коллективного пользования – серверам, которые помимо достаточной производительности должны обладать высокой надёжностью. В главе описаны способы повышения надёжности сервера.

В **третьей главе** описываются лини связи, используемые для передачи информации в сетях передачи данных. Вначале главы даётся описание эквивалентной электрической схемы лени связи, на основании которой выявляются основные характеристики линии связи и факторы, влияющие на качество передаваемого сигнала. Далее описываются широко используемые в настоящее время линии связи, как проводные, так и беспроводные. В конце главы описаны топологии, позволяющие объединить несколько компьютеров в сеть.

Четвёртая глава посвящена описанию методов кодирования и передачи данных по сетям. В главе приведены наиболее часто используемые методы кодирования, объясняются их достоинства и недостатки, приведены методам обнаружения и коррекции ошибок. Особое место уделено принципам сжатия информации и почему для разного рода информации используются разные методы сжатия.

В **пятой главе** описываются физические и технические решения, положенные в основу самой широко распространенной в России технологии Ethernet. Определяются основные характеристики данной технологии. Рассматриваются все существующие на данный момент разновидности данной технологии.

Шестая глава посвящена коммутаторам. В этой главе последовательно, по-

для их преодоления. Поскольку каждый порт коммутатора является процессором, то разработчики решили ввести в коммутатор дополнительные функции. В данной главе описываются только две дополнительные функции: идентификация источника и его подавление.

В **седьмой главе** очень подробно описывается метод создания отказоустойчивой сети на уровне коммутаторов с использованием избыточных каналов связи – метод связующего дерева – STP, его быстрый вариант – RSTP. а так же метод агрегации каналов – Ether Channel.

Восьмая глава посвящена виртуальным сетям. В этой главе объясняется необходимость использования виртуальных сетей и способы их организации на коммутаторах и протоколах. Объясняется целесообразность применения иерархического подхода к проектированию сетей.

В **девятой главе** описывается необходимость использования частных виртуальных сетей и способы их организации. В конце главы описан пример виртуального удалённого соединения через частную виртуальную сеть.

Десятая глава посвящена системе адресации, принятой в IP-сетях. В этой главе описывается необходимость применения другой системы адресации и подробно описывается система адресации в IP-сетях. Описывается пакет протокола IP, недостатки системы адресации и способы её преодоления в виде масок постоянной и переменной длины. Даётся понятие метода скользящего окна. Описывается структура пакета IPv6.

В **одиннадцатой главе** описывается назначение маршрутизатора, как устройства способного выбрать оптимальный путь прохождения пакета по составной сети с несколькими возможными путями. Даётся понятие авто-номной системы. Для составления таблиц маршрутизации маршрутизаторы используют разные методы сбора информации. Эти методы подробно описаны в данной главе. В данной главе подробно описана работа маршрутизирующего коммутатора – коммутатора третьего уровня.

Двенадцатая глава посвящена описанию протоколов TCP и UDP. Поскольку различие протоколов IP и TCP для многих не специалистов в области сетей кажется не очень большим, то возникает большая путаница между ними. Поэтому в данной главе подробно описывается назначение и принцип работы протокола TCP – как протокола гарантированной доставки сообщений. В тоже время для некоторых приложений и устройств не требуется гарантированная доставка сообщений, а требуется высокая производительность протокола. Специально для них был разработан протокол UDP.

В **главе тринадцать** описываются способы создания отказоустойчивых ка-

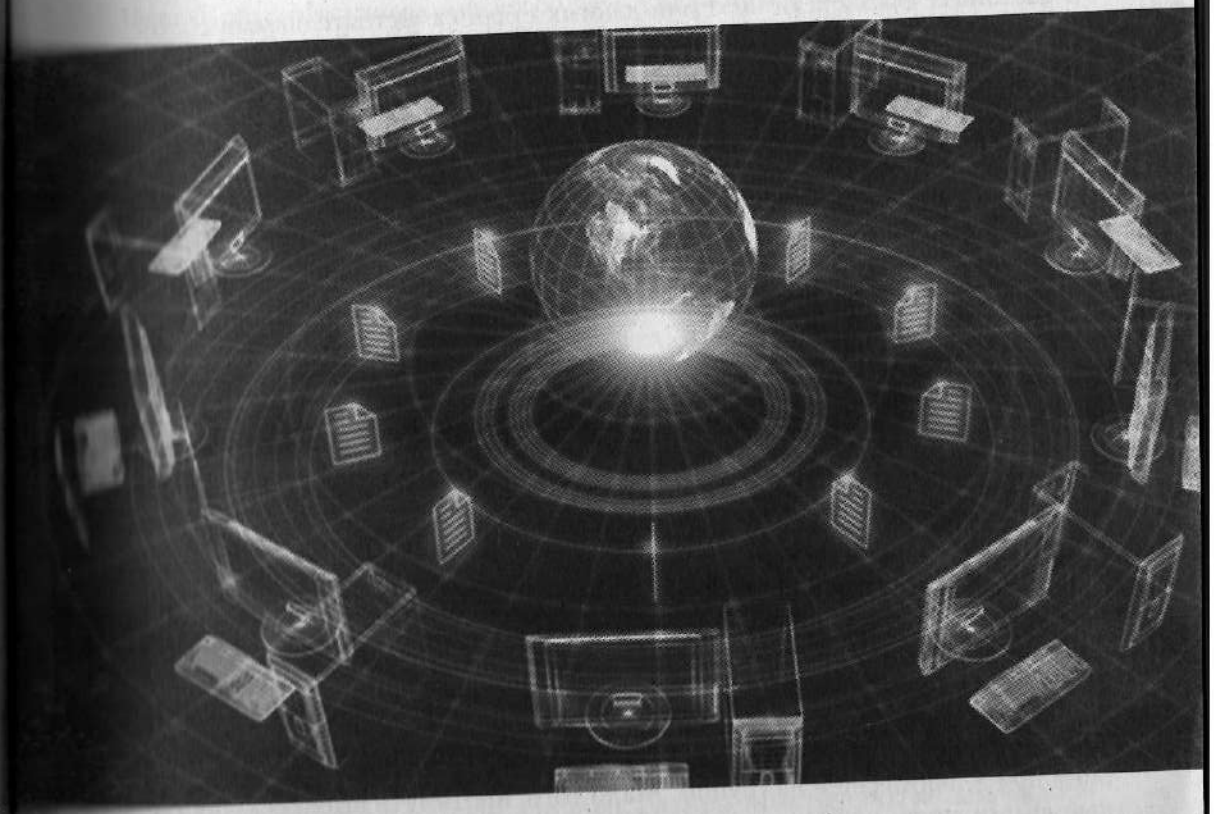
ие физического резервирования маршрутизаторов с использованием соот-
ветствующих протоколов.

Глава четырнадцать посвящена беспроводным сетям передачи дан-ных. В
этой главе в хронологическом порядке описаны разные технологии, исполь-
зуемые в разное время для беспроводной передачи информации. Подроб-
но описаны современные технологии – Bluetooth, Wi-Fi, WiMAX и сети мо-
бильной (сотовой) связи.

Автор прекрасно понимает, что некоторые важные вопросы, относя-щиеся к
сетям передачи данных, освящены или не в полной мере или вообще не вош-
ли в данную книгу. В первую очередь это относится к сетевым операцион-
ным системам. Включение материала по сетевым операционным системам в
данную книгу привело бы к её чрезмерному росту, кроме того многим чита-
телям этот материал не представлял бы большого интереса, если их интере-
сует только транспортная инфраструктура сети. Поэтому автор ограничился
описанием наиболее важных тем, без которых понимание работы сетей пере-
дачи данных вообще невозможно.

Глава 1.

ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ СЕТИ



Как отмечалось во введении, первыми появились глобальные сети для передачи информации от РЛС в центры управления, а затем на стрельбовые комплексы. Поскольку информация на РЛС, в центрах управления и стрельбовых комплексах подвергалась обработке, то основной функцией системы была обработка информации, а каналы передачи информации обеспечивали только её доставку. Такие территориально разнесённые системы обработки принято было называть вычислительными сетями. В вычислительных сетях каналы передачи информации рассматривались не как самостоятельные системы, а как часть многомашинного комплекса. В зависимости от скорости передачи каналы подразделялись на высокоскоростные – селекторные и низкоскоростные – мультиплексные.

В настоящее время сети передачи данных стали самостоятельными системами, обеспечивающими связь между клиентами (узлами) сети, поэтому их стали называть сетями передачи данных. Все компьютерные сети передачи данных в зависимости от протяженности линий связи и количества предоставляемых услуг принято делить на три большие группы:

- глобальные сети;
- локальные сети;
- сети студенческих городков, городские сети.

1.1. Глобальные сети

На первый взгляд кажется оправданным делить сети по размеру занимаемой ими территории, и чем больше территория, тем больше сеть. Поэтому глобальные сети должны и охватывать глобальные территории. Однако это понятие относительное. Например, национальные сети передачи данных Великобритании и Франции будет считаться глобальными сетями. А сеть передачи данных Красноярского края – локальной сетью Российской Федерации. Хотя сеть Красноярского края будет охватывать территорию в 11,5 раз большую, чем национальная сеть передачи данных Великобритании, или в 5 раз большую, чем национальная сеть передачи данных Франции. Таким образом, размер территории, охватываемой сетью, не является основным критерием при делении сетей на группы. Мало того, протянуть высокоскоростные линии связи на больших пространствах – проблема крайне дорогостоящая. Вследствие этого при делении сетей на

территория и скорость передачи информации отступают на второй план, а на первый план выступает критерий количества предоставляемых услуг. Качество услуги остаётся на совести того, кто её предоставляет.

Поскольку в большой сети количество узлов, предоставляющих услуги, велико, то и количество услуг будет велико. Таким образом, глобальной сетью можно считать сеть, которая пользователям сети предоставляет большое (не ограниченное) количество услуг при приемлемой скорости передачи для конкретного узла.

В этом месте на скорости передачи следует остановиться более подробно. Многие компании, предоставляющие услуги по передаче данных, гордо именующие себя провайдерами (provider – поставщик), в рекламных сообщениях обещают высочайшие скорости, например 1, 10 и даже 100 Мбит/с и более. Но они все лукавят, если не сказать более грубо. Рассмотрим небольшой пример. Две столицы нашей Родины: южная – Москва и северная – Санкт Петербург. В Москве проживает более 8 миллионов человек, а в Санкт Петербурге – более 5 миллионов. Если предположить, что в каждой из столиц всего по 2 миллиона человек имеют выход в сеть и между ними проложен высокоскоростной оптический кабель со скоростью в 10 Гбит/с (что маловероятно), то каждый из жителей получит не более 2,5 Кбит/с. Даже если учесть, что не все одновременно будут пользоваться сетью, а всего лишь каждый сотый, то и в этом случае скорость не превысит 250 Кбит/с, а уж 1 Мбит/с – несбыточная мечта. Мне могут возразить, мол, не все и не всё время передают информацию, поскольку после получения её некоторое время осмысливают, а линия в это время освобождается. Всё верно, но в часы "пик" скорость падает катастрофически. Так ведь это ещё при условии, что связь между столицами в 10 Гбит/с. Прокладка ещё одного кабеля принципиально не изменит ситуацию, а обойдётся в копеечку. Из всего сказанного следует, что в глобальных сетях скорость передачи информации невелика.

1.2. Локальные сети

В начале 60-х годов по мере удешевления процессоров появились новые способы организации вычислительного процесса. Начали развиваться интерактивные многотерминальные системы разделения времени. В таких системах с компьютером могли работать сразу несколько пользователей, каждый из которых получал в свое распоряжение терминал и таким образом мог вести диалог с компьютером (рис. 1.1). Время реакции вычислительной системы было настолько мало, что пользователь практически не замечал параллельной работы с компьютером других пользователей. Но при этом мог

у него создавалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу практически в любой момент и сразу же получить результат.

Таким образом, многотерминальные системы, работающие в режиме разделения времени, стали первым шагом на пути создания локальных вычислительных сетей. В этот период был справедлив так называемый **«закон Гроша», который эмпирически отражал уровень развития технологии того времени: производительность компьютера была пропорциональна квадрату его стоимости**, отсюда следовало, что за одну и ту же сумму было выгоднее купить одну мощную машину, чем две менее мощные, – суммарная мощность последних оказывалась намного ниже мощности дорогой машины.

В начале 70-х годов произошел технологический прорыв в области производства компьютерных компонентов – появились большие интегральные схемы. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами больших ЭВМ. Закон Гроша перестал соответствовать действительности, так как десяток мини-компьютеров выполнял некоторые задачи (как правило, хорошо распараллеливаемые) быстрее одной большой машины, а стоимость такой системы была ниже.

С появлением в 1981 году персональных ЭВМ ситуация резко изменилась. Стоимость первых персональных ЭВМ была невелика, однако сравнима с стоимостью жёсткого диска, поэтому компании покупали один жёсткий диск для нескольких машин. Пользователи старшего поколения помнят все прелести обмена информацией с помощью дискет – гибких дисков. Запись информации с первого раза на дискету считалась удачей, но она не гарантировала, что на приводе другой машины дискета прочитается. Одним словом, обмен информацией между машинами был длительным и творческим, сопровождаемым бесконечным количеством эмоций. Такое состояние с обменом информацией не могло продолжаться долго, в результате были созданы недорогие устройства обмена информацией.

Поскольку персональные ЭВМ имели крайне малые габариты, их ещё называли микромашинами, то на небольшой территории (локально) их можно было установить несколько штук, например в кабинете или зале. Персональные машины, связанные между собой линиями передачи данных, стали называть локальной сетью.

К сожалению, персональные ЭВМ пришли в нашу страну из-за рубежа и назывались они Personal Computer (персональный вычислитель). Но отечественному пользователю, разбе...

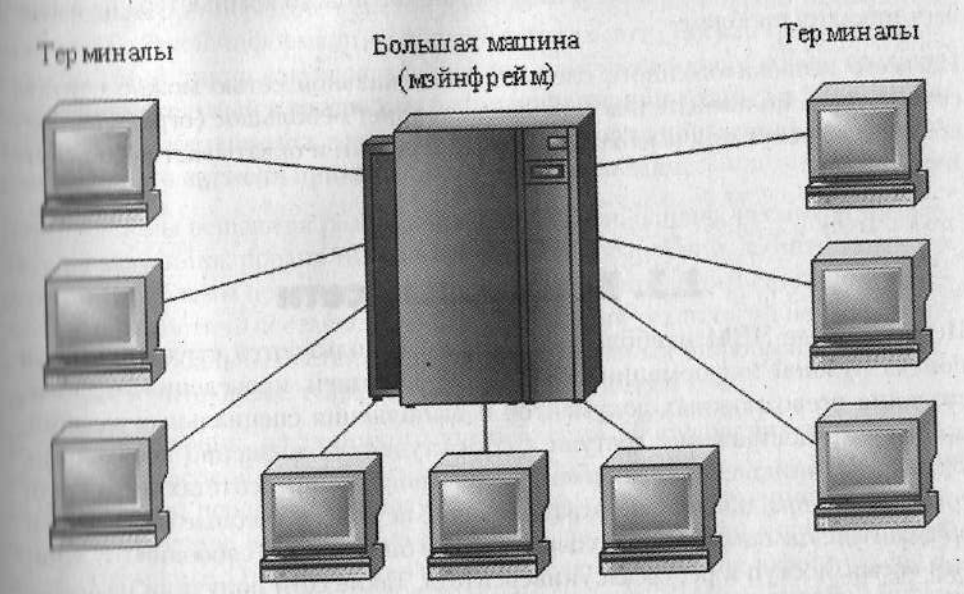


Рис. 1.1. ***Многотерминальная система разделения времени***

к вычислительной технике, слово «вычислитель» резало слух, то ли дело «КОМПЬЮТЕР», красиво и, самое главное, малопонятно. В кратчайшие сроки появилось огромное количество литературы, в которой персональный вычислитель был назван персональным компьютером. Это название быстро прижилось, особенно у молодого поколения (ведь их так учили в школе), поэтому все персональные вычислители мы в дальнейшем будем именовать персональными компьютерами.

Локальная сеть имеет в своём составе небольшое количество компьютеров, поэтому количество услуг, предоставляемых пользователям, ограничено. Но, с другой стороны, небольшая территория позволяет иметь короткие линии связи, поэтому их можно сделать крайне быстроходными при приемлемой цене. Первые локальные сети имели скорость передачи информации 2 Мбит/с.

Стандартные сетевые технологии превратили процесс построения локальной сети из искусства в рутинную работу. Для создания сети достаточно приобрести сетевые адаптеры соответствующего стандарта, стандартный кабель, присоединить адаптеры к кабелю стандартными разъемами и уста... компьютер одну из популярных сетевых операционных систем,

этого сеть начинает работать и подсоединение нового компьютера не вызывает никаких проблем.

Из всего вышеизложенного следует, что локальной сетью можно считать сеть, которая пользователям сети предоставляет небольшое (ограниченное) количество услуг при высокой скорости передачи и охватывает небольшую территорию.

1.3. Кампусные сети

Персональные ЭВМ наиболее интенсивно используются студентами: для поиска нужной информации, обмена информацией, проведения расчётов, создания всевозможных документов и выполнения специальных заданий, поэтому предоставление доступа к сети студентам жизненно необходимо. В связи с возникшей необходимостью во многих университетах мира стали создаваться локальные сети, охватывающие не только учебные классы, но и общежития, где проживают студенты, чтобы они имели, в свободное от занятий время, доступ к ресурсам университета. Такие сети получили название **кампусные сети** или **сети студенческих городков** (campus network).

Сети студенческого городка представляют нечто среднее между глобальными и локальными сетями. С точки зрения предоставляемых каналов связи они ближе к локальным сетям, поскольку на небольшой территории можно применять довольно быстроходные каналы связи. С точки зрения предоставления услуг они ближе к глобальным сетям, поскольку каждое подразделение в университете предоставляет свою информацию, например деканаты, кафедры, лаборатории, библиотеки, ректорат, приёмная комиссия и так далее; в результате количество предоставляемых услуг довольно велико.

В настоящее время сети передачи данных продолжают развиваться, причем достаточно быстро. Разрыв в скорости передачи информации между локальными и глобальными сетями постоянно сокращается во многом из-за появления высокоскоростных территориальных каналов связи, не уступающих по качеству кабельным системам локальных сетей. В глобальных сетях появляются службы доступа к ресурсам, такие же удобные и прозрачные, как и службы локальных сетей. Примером тому может служить самая популярная глобальная сеть Internet.

В настоящее время в сетях передачи стала присутствовать несвойственная им ранее информация – звуковая, видеоизображения, рисунки. Это потребовало внесения изменений в работу протоколов, сетевых операционных систем и коммуникационного оборудования. Сложность передачи такой ин-

к задержкам при передаче пакетов данных – задержки обычно приводят к искажению такой информации в конечных узлах сети. Так как традиционные службы сетей, такие как передача файлов или электронная почта, создают малочувствительный к задержкам поток информации (traffic) и все элементы сетей разрабатывались в расчёте на него, то появление потоков информации реального времени привело к большим проблемам.

Эти проблемы решаются различными способами, однако, несмотря на значительные усилия, предпринимаемые в этом направлении, до оптимального решения проблемы пока далеко, и в этой области предстоит ещё многое сделать, чтобы достичь поставленной цели – слияния технологий не только локальных и глобальных сетей, но и технологий любых информационных сетей – вычислительных, телефонных, телевизионных и т. п.

В последнее время, из-за резкого удешевления оборудования для построения сетей высокой производительности, стало возможным по не дорогим каналам связи передавать наряду с традиционной информацией – файлами и разнородную информацию – рисунки, звуковую и видеоинформацию в цифровом виде. Это позволило пользователю получить телефонную связь, возможность обмена рисунками, файлами и видеоинформацией. Многие компании стали протягивать такие линии связи в жилые дома и подключать квартиры. Такие сети стали называть городскими.

Основное отличие городских сетей от локальных и глобальных заключается в предоставлении комплекса услуг, в том числе кабельного телевидения, телефонной связи в цифровом виде и выхода в глобальные сети.

Компьютерные сети позволяют пользователям снизить затраты на другие виды связи, однако вопросы обеспечения безопасности в сетях передачи данных решаются в значительной степени путём усложнения как программного, так и аппаратного обеспечения. В некоторых случаях, когда безопасность особенно важна, от использования сети лучше вообще отказаться.

При создании вычислительных сетей их разработчикам пришлось решить много проблем. Далее будут рассмотрены только наиболее важные из них, причем в той последовательности, в которой они естественно возникали в процессе развития и совершенствования сетевых технологий

1.4. Обмен информацией ЭВМ с периферийным устройством

Для обмена данными между компьютером и периферийным устройством

собой набор проводов, соединяющих компьютер и ПУ, а также набор правил обмена информацией по этим проводам. Интерфейс реализуется со стороны компьютера совокупностью аппаратных и программных средств: контроллером ПУ и специальной программой, управляющей этим контроллером, которую часто называют драйвером соответствующего периферийного устройства.

Со стороны ПУ интерфейс чаще всего реализуется аппаратным устройством управления, хотя встречаются и программно-управляемые периферийные устройства. Периферийные устройства могут принимать от компьютера как данные, так и команды управления, в ответ на которые ПУ может выполнить специальные действия. Периферийное устройство использует интерфейс не только для приёма информации, но и для передачи информации в компьютер, например для передачи компьютеру данных о своём состоянии.

Поскольку в передаче информации принимает участие большое количество как программных, так и аппаратных средств разного уровня, то для лучшего понимания всех нюансов работы такого программно-аппаратного комплекса целесообразно рассмотреть пример.

Итак, у нас имеется компьютер и лазерное печатающее устройство, соединённые между собой соответствующим кабелем (рис. 1.2). Поскольку по кабелю передаются электрические сигналы, то для преобразования их в биты, байты и обратно в компьютере и ПУ установлены преобразователи, которые выполняют ещё и некоторые дополнительные функции управления, связанные с организацией обмена. Такое устройство в англоязычной литературе называется **controller (контролёр).** Однако контролёр в сознании русского человека ассоциируется с пожилым человеком, частенько со сварливой бабулей, проверяющим и компостирующим билеты в общественном транспорте – в автобусе, трамвае, троллейбусе и т.п. Но нельзя же сравнивать такое высокоинтеллектуальное устройство вычислительной машины с обычным контролёром. К тому же неудержимая тяга нашего человека ко всякого рода иностранным словам привела к тому, что в слове контролёр ударение переехало на второй слог и данное устройство стало гордо именоваться контроллером.

Каждый котроллер в своём составе помимо устройства управления содержит внешний порт ввода/вывода информации, к которому собственно и подключается соединительный кабель. Контроллер компьютера содержит ещё и внутренний порт для обмена информацией с процессором. Для работы компьютера на нём установлена операционная система. Пользователем на компьютере запущено приложение, допустим, текстовый редактор типа Word. Пользователь в редакторе создал документ и [illegible]

указателем "мышки" по пиктограмме "принтер". Здесь следует заметить, что слово принтер является побуквенным отображением слова printer – опять подвела тяга к иностранным словам. А далее происходят следующие удивительные действия.

Приложение в общих чертах знает, в каком виде необходимо подготовить документ для того, чтобы его понял принтер. Но оно не знает всех тонкостей работы данного конкретного принтера. Поскольку разработчики принтера лучше всех знают, как работает принтер, и в каком виде ему нужно подготавливать документы, то они вместе с принтером поставляют специальную программу работы с ним, которая в англоязычной литературе называется driver, что в переводе на русский язык означает: шофёр, погонщик, надсмотрщик, приспособление. В данном случае суть программы вернее всего отражает – надсмотрщик, но назвать программу «надсмотрщиком» у программистов рука не поднялась, поскольку у большинства из старшего поколения слово «надсмотрщик» ассоциируется с тюремным персоналом. Следует отметить, что в вычислительной технике очень много заимствованных слов, к сожалению, большинство из них употребляются абсолютно бездумно.

Приложение напрямую не может обратиться к принтеру, потому что не знает, как с ним работать. Однако на компьютере установлена операционная система (ОС), которая призвана облегчать работу всех приложений, поэтому приложение обращается к ОС за помощью (действие 1). На рис. 1.2 все обращения и передачи показаны стрелочками, а последовательность действий обозначена цифрами. Операционная система воспринимает запрос от приложения и определяет, что ей необходимо оказать услуги по работе с лазерным принтером определённой марки. Поскольку ОС тоже не владеет тонкостями работы с лазерным принтером она, порывшись в своих закромах, находит и запускает программу – драйвер данного принтера – 2, которому сообщает, что в таком-то диапазоне адресов оперативной памяти находится документ, который необходимо вывести на печать. Драйвер указанный документ переводит в термины, понятные принтеру, снабжая его изрядным количеством пометок (служебной информацией), и начинает передавать его во внутренний порт контроллера – 3.

Контроллер совершенно не разбирается в передаваемой информации, его задача её просто передать. Контроллер имеет небольшую внутреннюю память, поэтому получает от драйвера информацию частями, обычно словами (слово, в зависимости от процессора, может содержать от 1 до 8 байт), которую преобразует в электрические сигналы требуемого вида, снабжает дополнительной информацией, необходимой для организации обмена, и посылает на порт входа/выхода. С порта входа/выхода сигналы по кабелю поступают в

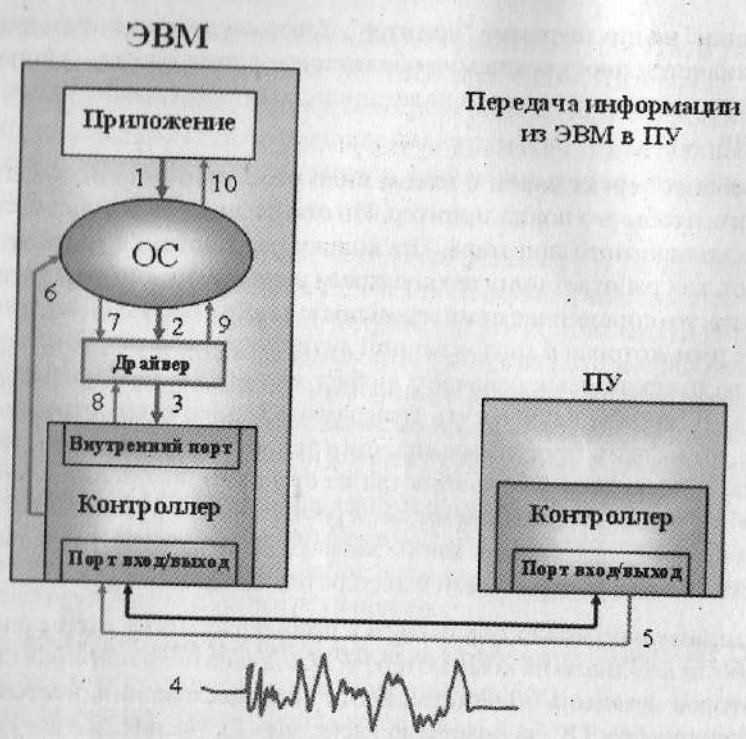


Рис. 1.2. ***Связь компьютера с периферийным устройством***

Контроллер принтера получает информацию со своего порта входа/выхода, преобразует её в слова (байты), расшифровывает и понимает, что ему необходимо будет напечатать документ с требуемым качеством, определённым шрифтом. Он запускает механизм печати, который включает печку, после прогрева печки запускается механизм подачи бумаги из лотка. И тут возникает вполне тривиальная ситуация – лист бумаги заминается.

Контроллеру принтера дальше нет смысла получать информацию из компьютера, поскольку он её всё равно напечатать не сможет, поэтому котроллер принтера генерирует сообщение компьютеру и посылает его через свой порт входа/выхода на порт входа/выхода контроллера компьютера – 5.

Котроллер компьютера получает информацию от принтера, преобразовывает её в слова, но он не знает, что с ней делать, поэтому обращается к операционной системе за помощью – 6. Операционная система, находясь к полудрёме, поскольку все процессы запущены и работают нормально, встрепену-

и поэтому обратилась к драйверу, который уже запущен – 7, с «просьбой» принять информацию из входного порта контроллера, поскольку она в ней ничего не понимает.

Драйвер принтера, получив информацию от контроллера – 8, преобразует её в термины понятные приложению, однако напрямую к приложению он обращаться не может, поэтому просит операционную систему о помощи – 9. Операционная система сообщает драйверу диапазон адресов, куда необходимо поместить полученную информацию, и одновременно извещает приложение – 10 о получении для него сообщения.

Приложение, получив эту информацию, генерирует сообщение для пользователя. В результате чего на экране появляется окно с текстом "бумага замята". Пользователь подходит к принтеру, удаляет замятый лист бумаги и переводит принтер в рабочее состояние. О чём принтер немедленно извещает приложение, и процесс печати продолжается с прерванного места. Через некоторое время, скорее всего, документ будет полностью напечатан.

1.5. Взаимодействие двух компьютеров

Поскольку процесс передачи информации из компьютера в котроллер устройства решён, то почему бы не попробовать передать информацию из компьютера в контроллер другого компьютера. В самом простом случае взаимодействие компьютеров может быть реализовано с помощью тех же самых средств, которые используются для взаимодействия компьютера с периферией.

В отличие от взаимодействия компьютера с периферийным устройством, когда приложение работает, как правило, только с одной стороны – со стороны компьютера, в сетях происходит взаимодействие двух приложений, работающих на каждом из компьютеров.

Приложение, работающее на одном компьютере, не может получить непосредственный доступ к ресурсам другого компьютера – его дискам, файлам, принтеру. Оно может только «попросить» об этом приложение, работающее на том компьютере, которому принадлежат эти ресурсы. Эти «просьбы» выражаются в виде сообщений, передаваемых по каналам связи между компьютерами. Сообщения могут содержать не только команды на выполнение некоторых действий, но и собственно информацию – содержимое передаваемого файла.

Рассмотрим случай, когда пользователь, работающий в приложении на пер-

созданный им документ на жёстком диске второй ЭВМ (рис. 1.3). Эти компьютеры связаны между собой кабелем, подключённым к портам входа/выхода контроллеров соответствующих компьютеров. На ЭВМ установлены операционные системы.

После того, как пользователь дал команду редактору текстов "сохранить файл как", указав в качестве места хранения жёсткий диск другой ЭВМ, приложение обращается за помощью к операционной системе – 1, поскольку оно не в состоянии самостоятельно передать информацию в другую ЭВМ.

Операционная система определяет, что поступивший запрос на сохранение относится не к локальным (внутренним) ресурсам, а к внешним, поэтому она должна передать исходный файл с помощью соответствующего своего контроллера на вход контроллера другого компьютера. Операционная система не может это сделать напрямую, поскольку не знает всех тонкостей работы котроллера. Для работы с контроллером ОС загружает драйвер котроллера и сообщает ему диапазон адресов оперативной памяти, где располагается документ, подлежащий сохранению – 2.

Драйвер котроллера находит документ в памяти, преобразует его в вид, необходимый для передачи на другой компьютер, и начинает порциями передавать информацию на внутренний порт контроллера. Контроллер преобразует полученную информацию в электрические сигналы, снабжает её некоторым количеством служебной информации в соответствии с правилами, принятыми для данного типа соединения, и передаёт на порт входа/выхода. С порта входа/выхода сигналы по кабелю поступают на порт входа/выхода другого компьютера – 4.

На ЭВМ 2 работает своё приложение, и никто не подозревает, что первая ЭВМ начала передачу. Однако контроллер 1 второго компьютера начинает получать информацию от первого компьютера. Поскольку контроллер 1 не разбирается в поступившей информации (он может только сказать, что она поступила без искажений), он обращается за услугами к операционной системе, посылая ей сигнал прерывания – 5.

Операционная система определяет, что к ней обратился контроллер 1, и для работы с ним она загружает драйвер 1 и просит его разобраться с пришедшей информацией – 6. Драйвер 1 получает информацию с внутреннего порта своего контроллера – 7, преобразовывает её в термины понятные ОС и передаёт операционной системе – 8.

Операционная система, получив сообщение, определяет, что компьютер 1 желает сохранить информацию на жёстком диске компьютера 2. Но жёсткий диск – устройство крайне сложное, и операционная система не умеет

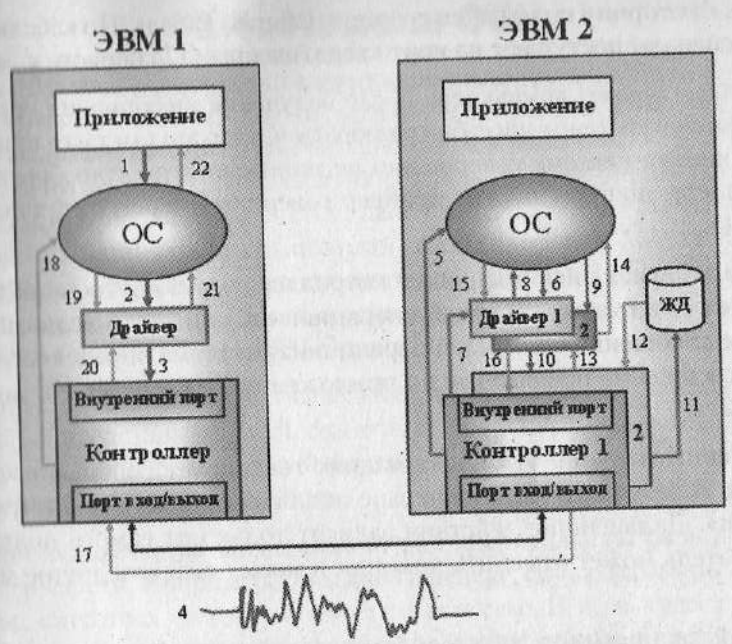


Рис. 1.3. ***Взаимодействие двух компьютеров***

им управлять, поэтому она загружает соответствующий драйвер жёсткого диска 2 и просит его выполнить операцию сохранения информации – 9, поступающей от драйвера 1.

Драйвер жёсткого диска обращается к контроллеру жёсткого диска, который умеет генерировать электрические сигналы управления жёстким диском – 10, и передаёт ему первую порцию полученной информации. Контроллер жёсткого диска обращается к жёсткому диску и осуществляет проверку доступного пространства – 11. И вот незадача, в выбранном разделе жёсткого диска осталось мало места и весь файл не может быть сохранён – 12.

Контроллер жёсткого диска 2 сообщает эту прискорбную информацию драйверу жёсткого диска – 13. Драйвер жёсткого диска 2 переводит эту информацию на язык понятный операционной системе и передаёт её ОС – 14. Операционная система определяет, что полученную информацию необходимо передать в другой компьютер, поэтому она её транслирует соответствующему драйверу 1 – 15. Драйвер 1 переводит полученное сообщение в язык понятный котроллеру 1 и передаёт его (16) на внутренний порт контроллера 1.

Контроллер 1 генерирует последовательность электрических импульсов, в соответствии с полученной информацией, и посылает их на порт входа/

выхода, к которому подключен соединительный кабель. По кабелю электрические сигналы поступают на порт входа/выхода (17) первого компьютера.

Контроллер первого компьютера после получения информации обращается к операционной системе с просьбой разобраться, а что это там такое пришло – 18? Операционная система совершенно не понимает этот «тарабарский язык» контроллера, поэтому просит драйвер совершить перевод на понятный ей язык – 19.

Драйвер забирает информацию из котроллера через внутренний порт – 20, переводит её на язык понятный операционной системе и возвращает переведённое сообщение ОС – 21. Операционная система определяет, что полученное сообщение предназначено приложению, и передаёт его по назначению – 22.

Приложение, получив эту информацию, генерирует сообщение для пользователя. В результате чего на экране появляется окно с текстом «диск переполнен». Дальнейшие действия зависят только от самого пользователя. Пользователь может отменить команду, а может выбрать другое место хранения.

Как видно из примеров, казалось бы, простейшие действия – печать документа или его сохранение – приводят к выполнению огромного количества разнообразных действий, начиная с самых верхних уровней программного обеспечения до аппаратных средств по генерации электрических сигналов и передачи их по линии связи.

Для соединения компьютеров желательно иметь стандартные средства сопряжения. Но для разработки программного или аппаратного обеспечения, реализующего тот или иной процесс, сначала необходимо составить модель процесса. Поскольку, как было показано выше, процесс передачи информации между компьютерами весьма сложен, то и модель должна быть не менее сложна.

Во все времена для упрощения решения крайне сложной задачи применяется метод декомпозиции, то есть разбиения задачи на относительно самостоятельные и более простые подзадачи. Декомпозиция позволяет вполне чётко сформулировать условия и методы решения каждой подзадачи и определить взаимосвязь между ними. Причём взаимодействие подзадач должно определяться небольшим набором простых правил, а количество взаимодействий необходимо свести к минимуму. В идеале одна подзадача должна взаимодействовать не более чем с двумя-тремя подзадачами. Процесс передачи информации между компьютерами не стал исключением.

1.6. Модель OSI

В начале 80-х годов несколько международных организаций по стандартизации совместными усилиями разработали модель, которую назвали моделью взаимодействия открытых систем или моделью OSI. Эта модель сыграла значительную роль в развитии сетей, поскольку определяет различные уровни взаимодействия систем, даёт им стандартные имена и определяет, какие функции должен выполнять каждый уровень. Модель OSI была разработана на основании практического опыта, полученного при создании глобальных компьютерных сетей, в основном, в 70-е годы. Полное описание этой модели занимает более 1000 страниц текста.

В модели OSI (рис. 1.4) средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело только с одним аспектом взаимодействия сетевых устройств.

В истории человечества число семь играет значительную роль и является почти магическим. Например, семь дней в неделе, семь раз отмерь – один отрежь, семь смертных грехов, семья – в смысле семь Я, семь чудес света и так далее. Очевидно, отдавая дань традициям, разработчики и разделили весь процесс передачи информации между компьютерами на семь частей.

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей.

Собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень. Начнём рассмотрение уровней модели с самого нижнего – физического.

- **Физический уровень** отвечает за передачу информации по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или радиоканал. На этом же уровне определяются характеристики электрических сигналов, для передачи цифровой информации (крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов, временные параметры импульсов, интервалы следования и т.д.). Кроме того, этот уровень определяет типы разъемов и назначение каждого контакта. Функции физического уровня реализуются во всех устройствах, подключённых к сети. Со стороны компьютера функции физического уровня выполняются специальным устройством – сетевым адаптером

- **Канальный уровень** обеспечивает проверку доступности среды передачи данных и корректность передачи информации. Для этого на канальном уровне биты группируются в наборы, называемые кадрами. Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, обрабатывая все байты кадра определённым способом, а также вычисляя контрольную сумму и добавляя её к кадру. Для доставки кадров канальный уровень использует физические адреса устройств, он это делает только в сети, для которой был разработан. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

  Впоследствии канальный уровень был разделён на два подуровня: подуровень управления логической связью (LLC) и подуровень управления доступом к среде (MAC).

  Подуровень **управления доступом к среде** – нижний из двух подуровней. Он обеспечивает совместный доступ к физическому уровню. Этот подуровень напрямую связан с сетевым адаптером и отвечает за бесперебойную передачу данных между двумя компьютерами сети.

  Подуровень **управления логической связью** устанавливает канал связи и определяет использование логических точек интерфейса, называемых точками доступа к услугам. Другие компьютеры, ссылаясь на точки доступа к услугам, могут передавать информацию с этого подуровня на верхние уровни модели OSI.

- **Сетевой уровень** служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать различные принципы передачи сообщений между узлами и обладать произвольной структурой связей. Для передачи информации между сетями сетевой уровень использует свою собственную систему адресации.

  При организации доставки пакетов на сетевом уровне используется понятие «номер сети». В этом случае адрес получателя состоит из старшей части – номера сети и младшей – номера узла в этой сети. Все узлы одной сети должны иметь одну и ту же старшую часть адреса.

  Проблема выбора наилучшего пути является одной из главных задач сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени.

  Сетевой уровень также решает задачи согласования разных технологий, упрощая адресацию в крупных сетях и создавая надежные и гибкие барьеры на пути нежелательного трафика между сетями.

- **Транспортный уровень** обеспечивает верхним уровням передачу данных с той степенью надежности, которая им требуется, предоставляя пять классов сервиса. Эти виды сервиса отличаются качеством предоставляемых услуг: срочность, надёжность, возможность восстановления прерванной связи, наличие средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное – способность к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

  Выбор класса сервиса транспортного уровня зависит от того, в какой степени задача обеспечения надёжности решается самими приложениями и протоколами более высоких уровней, с одной стороны, и насколько надёжной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного, с другой стороны.

- **Сеансовый уровень** обеспечивает управление диалогом: определяет необходимые ресурсы для диалога и их доступность, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации, вставляя контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад, к последней контрольной точке, а не начинать процесс с самого начала. На этом уровне производится сбор всевозможных статистических данных, часть которых, в частности, используют операторы сотовой связи и Internet для взимания платы за свои услуги.

- **Представительный уровень** (уровень представления) имеет дело с формой представления передаваемой по сети информации, не меняя при этом её содержания. Этот уровень может изменять кодирование информации. Например, Вы получили сообщение и пытаетесь его открыть с помощью редактора текстов, а на экране при этом появляется последовательность квадратиков, кружочков и ромбиков (такое часто бывает при получении информации из Internet). Однако стоит лишь изменить систему кодирования (ASCII в EBCDIC), как полученная информация приобретает привычный текстовый вид. Таким образом, изменилась только форма представления, а не содержание. На этом уровне мо-

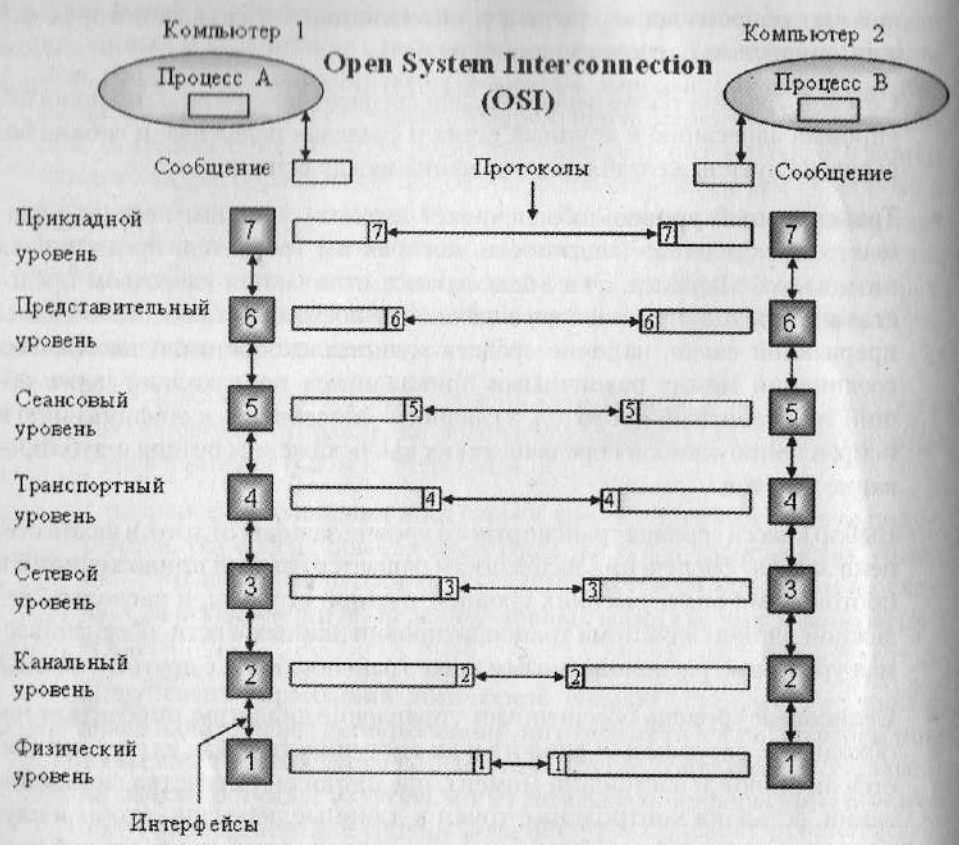


Рис. 1.4. ***Модель взаимодействия открытых систем OSI***

- **Прикладной уровень** – это просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые web-страницы, а также организуют свою совместную работу, например с помощью протокола электронной почты.

Три нижних уровня – физический, канальный и сетевой – являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием.

Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют какие бы то ни было изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних и наоборот. Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений.

На представленной на рис. 5 модели информация движется от верхнего уровня к нижним, а затем наоборот. Каждый уровень к полученной информации добавляет свою служебную информацию, не вникая в принятую. В результате, когда информация попадает на физический уровень, количество служебной информации может превышать количество информации, сгенерированной приложением.

Продемонстрируем сказанное на простом примере. Допустим, одна компания имеет договорные отношения с другой компанией. Директор одной компании в ответ на невыполнение в срок своих обязательств согласно договору с другой компанией решает написать директору той компании гневное письмо. Он быстренько в общих чертах набрасывает тезисы письма и с пометками передаёт его секретарю, чтобы он облёк его в литературную форму.

Секретарь, получив этот набросок, облачает его в литературную форму, не меняя содержания, регистрирует в журнале и передаёт его юристу, чтобы тот проверил правильность претензий с юридической точки зрения. Юрист проверяет письмо и, если требуется, снабжает его ссылками на соответствующие случаю статьи законов и передаёт в канцелярию.

Канцелярия, не вникая с суть написанного, регистрирует письмо, запечатывает его в конверт и с курьером отправляет на почту. На почте письмо принимают, выдают квитанцию и по прошествии некоторого времени отправляют на почтамт, где будет произведена сортировка.

На почтамте вся поступившая почта сортируется по направлениям и городам, запечатывается в большие мешки и оправляется по назначению. Причём маршрут следования будет выбираться на каждом этапе продвижения мешка. Поэтому два письма, отправленные с небольшим перерывом по одному адресу, могут пройти различный путь.

На каждом из этапов передачи сообщения все службы и люди добавляют к нему свою служебную информацию, необходимую только им. Это позволяет отследить путь сообщения и в случае его задержки или утраты всегда можно определить, где это произошло и кто виноват.

В пункте получения письма оно проходит описанный выше путь в обратном направлении.

Модель OSI, как это следует из её названия, описывает взаимосвязи открытых систем, открытость заключается в открытости спецификаций всем заинтересованным лицам. Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные

аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

1.7. Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Поскольку в идеале для каждого уровня модели OSI необходим свой протокол, то все протоколы в рамках одной идеологии объединяются в группы, называемые стеками. В настоящее время в сетях используется большое количество стеков коммуникационных протоколов. Наиболее популярными являются стеки: TCP/IP, IPX/SPX, NetBIOS /SMB, DECnet и OSI. Эти стеки на нижних уровнях (физическом и канальном) используют одни и те же хорошо стандартизованные протоколы Ethernet, Token Ring, FDDI и некоторые другие, которые позволяют использовать во всех сетях одну и ту же аппаратуру. Зато на верхних уровнях все стеки работают по своим собственным протоколам. Эти протоколы часто не соответствуют рекомендуемому моделью OSI разбиению на уровни. В частности, функции сеансового и представительного уровня, как правило, объединены с прикладным уровнем. Такое несоответствие связано с тем, что модель OSI появилась как результат обобщения уже существующих и реально используемых стеков, а не наоборот.

Стек OSI

Следует четко различать модель OSI и стек OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор вполне конкретных спецификаций протоколов. В отличие от других стеков протоколов стек OSI полностью соответствует модели OSI и включает спецификации протоколов для всех семи уровней взаимодействия, определённых в этой модели.

Протоколы стека OSI отличает большая сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все случаи жизни и все существующие и разрабатываемые технологии. К этому нужно ещё добавить и последствия большого количества политических компромиссов, неизбежных при принятии международных стандартов по такому злободневному вопросу, как построение открытых сетей.

Из-за своей сложности протоколы OSI требуют больших вычислительных затрат от центрального процессора, что делает их наиболее подходящими для мощных машин, а не для сетей персональных компьютеров.

Стек TCP/IP

Стек TCP/IP был разработан более 30 лет назад для связи компьютеров в рамках экспериментальной сети ARPANET, как набор общих протоколов для разнородной среды. Большой вклад в развитие стека TCP/IP внёс университет Беркли, реализовав протоколы стека в своей версии ОС UNIX. Популярность этой операционной системы привела к широкому распространению протоколов стека TCP/IP. Сегодня этот стек используется для связи компьютеров Всемирной информационной сети Internet, а также в огромном числе корпоративных сетей.

Основными протоколами стека, давшими ему название, являются протоколы IP и TCP. Эти протоколы согласно модели OSI относятся к сетевому и транспортному уровням соответственно: IP обеспечивает продвижение пакета по составной сети, а TCP гарантирует надёжность его доставки.

Сегодня стек TCP/IP представляет собой один из самых распространенных стеков транспортных протоколов сетей. Действительно, только в сети Internet объединено более 20 миллионов компьютеров по всему миру, которые взаимодействуют друг с другом с помощью стека протоколов TCP/IP.

Хотя протоколы TCP/IP неразрывно связаны с Internet на основе этого стека, существует большое количество локальных, корпоративных и территориальных сетей, непосредственно не являющихся частями Internet. Чтобы отличать их от Internet, эти сети называют просто IP-сетями.

<table>
<tr><th>OSI</th><th>NetWare</th><th>TCP/IP</th><th>IBM</th><th>DECnet</th></tr>
<tr><td>Прикладной</td><td rowspan="2">NCP</td><td>Telnet</td><td rowspan="2">SMB</td><td></td></tr>
<tr><td>Представления</td><td>SNNP, WWW</td><td></td></tr>
<tr><td>Сеансовый</td><td>SAO</td><td rowspan="2">TCP</td><td rowspan="3">NetBIOS</td><td></td></tr>
<tr><td>Транспортный</td><td>SPX</td><td></td></tr>
<tr><td>Сетевой</td><td>IPX</td><td>IP</td><td></td></tr>
<tr><td>Канальный</td><td colspan="4">Тип архитектуры</td></tr>
<tr><td>Физический</td><td colspan="4">Среда передачи</td></tr>
</table>

Рис. 1.5. ***Стеки протоколов***

Поскольку стек TCP/IP изначально создавался для глобальной сети Internet, он имеет много особенностей. В частности, он позволяет фрагментировать пакеты. Действительно, большая составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть установлена собственная величина максимальной длины единицы передаваемых данных. В таком случае при переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого сообщения на части. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью протоколов TCP/IP является гибкая система адресации, позволяющая более просто подключать к интерсети сети других технологий. В стеке TCP/IP очень экономно используются возможности широковещательных рассылок. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако за получаемые преимущества надо платить, и платой здесь оказываются высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации высоких вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети различных централизованных служб. Каждая из этих служб направлена на облегчение администрирования сети, в том числе и на облегчение конфигурирования оборудования, но в то же время сама требует пристального внимания со стороны администраторов.

Стек IPX/SPX

Этот стек является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare ещё в начале 80-х годов. Протоколы сетевого и сеансового уровней IPX и SPX, которые дали название стеку, являются прямой адаптацией протоколов XNS фирмы Xerox, распространённых в гораздо меньшей степени, чем стек IPX/SPX. Популярность стека IPX/SPX непосредственно связана с операционной системой NetWare, которая ещё сохраняет мировое лидерство по числу установленных систем.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare (до версии 4.0) на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти и небольшие вычислительные мощности. В результате протоколы стека IPX/SPX хорошо работают в локальных сетях и не очень в больших корпоративных сетях, поскольку слишком перегружают медленные глобальные каналы связи широковещательными пакетами, которые интенсивно используются несколькими протоколами этого стека. Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell и на его реализацию нужно получать лицензию, долгое время ограничивали распространенность его только сетями NetWare. Однако с момента выпуска версии NetWare 4.0 компания Novell внесла и продолжает вносить в свои протоколы серьезные изменения, направленные на их адаптацию для работы в корпоративных сетях. Сейчас стек IPX/ SPX реализован не только в NetWare, но и в нескольких других популярных сетевых ОС, например UNIX, Solaris, Windows NT.

Стек NetBIOS/SMB

Этот стек широко используется в продуктах компаний IBM и Microsoft. Протокол NetBIOS появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменён протоколом расширенного пользовательского интерфейса NetBEUI. Для обеспечения совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. Протокол NetBEUI разрабатывался как эффективный протокол, потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 узлов. Этот протокол содержит много полезных сетевых функций, которые можно отнести к сетевому, транспортному и сеансовому уровням модели OSI, однако с его помощью невозможна маршрутизация пакетов. Это ограничивает применение протокола NetBEUI локальными сетями, не разделёнными на подсети, и делает невозможным его использование в составных сетях.

Протокол SMB выполняет функции сеансового, представительного и прикладного уровней. На основе SMB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

Стек DECnet

Набор протоколов DECnet является собственным стеком протоколов фирмы DEC. Этот набор программно-аппаратных продуктов реализует архитектуру DNA (архитектура цифровых сетей). Указанная архитектура используется для построения сетей путём объединения локальных сетей, построенных на технологиях Ethernet, FDDI, и глобальных сетей, которые используют средства передачи конфиденциальных и общедоступных данных.

Набор протоколов DECnet допускает использование помимо собственных

протоколов ещё и протоколов стеков TCP/IP и OSI. Данные протоколы принадлежат к числу маршрутизируемых протоколов. Стек протоколов периодически обновляется, каждое обновление называется фазой. Последняя версия стека протоколов DECnet – DECnet Phase VI.

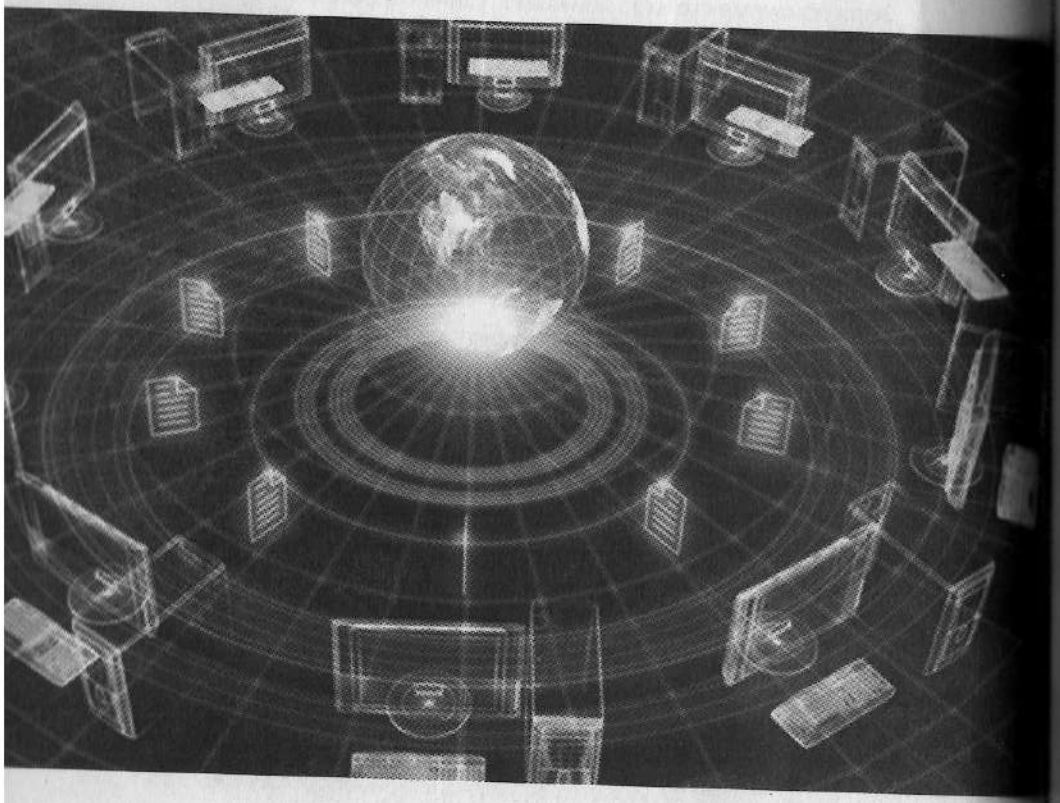
Контрольные вопросы

1. *Дайте определение локальной сети.*
2. *Дайте определение глобальных сетей.*
3. *Дайте определение сети студенческого городка.*
4. *Дайте определение городской сети.*
5. *Назовите основные отличия сети студенческого городка от локальной сети.*
6. *Назовите основные отличия глобальной сети от кампусной сети.*
7. *Назовите основные черты городской сети.*
8. *Чем отличается многотерминальная сеть от локальной сети?*
9. *Назовите основные причины возникновения стандартов на технологии локальных сетей.*
10. *Каковы основные тенденции в области развития компьютерных сетей?*
11. *Назовите основные компоненты сетей передачи данных.*
12. *Что описывает топология применительно к сети?*
13. *Перечислите основные достоинства и недостатки топологии общая шина.*
14. *Перечислите основные достоинства и недостатки топологии кольцо.*
15. *Перечислите основные достоинства и недостатки топологии звезда.*
16. *Перечислите основные достоинства и недостатки звездообразной топологии.*
17. *Перечислите основные достоинства и недостатки полносвязанной топологии.*
18. *Назовите назначение модели OSI.*
19. *Перечислите все уровни модели OSI.*
20. *Дайте определение понятия стек протоколов.*
21. *Назовите стандартные стеки протоколов.*
22. *Перечислите особенности стека протоколов TCP/IP.*
23. *Перечислите особенности стека протоколов IPX/SPX.*

24. *Перечислите особенности стека протоколов NetBIOS/SMB.*
25. *Перечислите основные требования, предъявляемые к вычислительным сетям.*

Глава 2.

КОМПОНЕНТЫ СЕТИ



Даже при беглом ознакомлении с работой сети передачи данных становится ясно, что сеть представляет собой большой и сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных средств. Изучение сети в целом предполагает знание принципов работы её составляющих частей. Основными компонентами сети передачи данных являются:

- рабочие станции;
- серверы;
- коммуникационное оборудование;
- сетевые операционные системы.

Весь дальнейший материал книги посвящён рассмотрению именно этих четырёх компонент.

2.1. Рабочие станции

В основе любой сети лежит аппаратный слой стандартизованных компьютерных платформ. В настоящее время в сетях широко применяются компьютеры различных классов – от персональных компьютеров до суперЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью. Любое устройство, подключённое к сети, называется узлом сети.

Рабочей станцией называется компьютер конечного пользователя, подключённый к сети. Со стороны сети к такому компьютеру предъявляется одно единственное требование – в нём должна быть установлена сетевая карта, с помощью которой компьютер может быть подключен к сети. Естественно, что для нормальной работы сетевой карты на компьютере должен быть запущен соответствующий драйвер.

Что же касается мощности компьютера, выполняющихся на нём приложений и других характеристик, то это прерогатива самого пользователя. В зависимости от типа решаемых задач пользователем, рабочие станции могут лежать в диапазоне от простейших маломощных компьютеров до мощнейших многопроцессорных (иногда многомашинных) сверхнадежных графических станций с несколькими мониторами и огромным дисковым пространством, снабжённым системами резервного копирования. Но для сети самое главное заключается в том, что они должны иметь средства подключения к сети. Всё остальное зависит только от финансовых возможностей пользователя и для сети не имеет никакого значения.

2.2. Серверы

Сервером называется компьютер, предоставляющий свои ресурсы в общее пользование. Именем ресурса, которым располагает сервер, принято называть и сам сервер. Например, если сервер предлагает в общее пользование своё дисковое пространство для хранения или получения файлов, то такой сервер называется файловым сервером. Сервер, предоставляющий печатающие устройства, называется сервером печати, а предоставляющий модемы – телекоммуникационным сервером соответственно, услуги электронной почты – почтовым сервером, услуги работы с базой данных – сервером баз данных, услуги по работе с приложением – сервером приложений и т.д. Таким образом, в сети может присутствовать большое количество серверов. Предоставление ресурса поддерживается соответствующим программным обеспечением, установленным на сервере. На одном компьютере одновременно может быть установлено несколько программных систем предоставления ресурсов, в этом случае компьютер будет выступать в роли нескольких серверов, например файлового и сервера печати.

Поскольку сервер является машиной коллективного пользования, то со стороны сети к нему предъявляется целый набор весьма специфических требований, а именно:

- достаточная производительность;
- большое дисковое пространство;
- достаточный объём оперативной памяти;
- высокая надёжность.

Рассмотрим все эти требования более подробно.

Достаточная производительность

Большинство пользователей заблуждаются в одном – они думают, раз это сервер, то он должен иметь очень высокую производительность. На самом деле это совершенно не так, всё зависит от типа ресурса, который он предоставляет в общее пользование. Так, файловый сервер, подключённый к сети Ethernet со скоростью передачи 100 Мбит/с, будет иметь эффективную скорость передачи не более 6 – 7 Мбайт/с. Современные самые "захудалые" жёсткие диски, имеющие интерфейс передачи данных Serial ATA -100, передают данные со скоростью в 100 Мбайт/с, что как минимум на порядок выше скорости передачи данных по сети. Поэтому с работой по поиску нужной информации на диске и передаче её по сети конкретному пользователю справится любой самый не производительный микропроцессор. Он быстро

найдёт нужную информацию, быстро прочитает её в память и медленно-медленно будет передавать в сеть, почти простаивая.

Другое дело сервер баз данных или приложений. По запросу пользователя о поиске нужной информации, сервер должен переворошить все свои закрома, найти нужную информацию, произвести какие-то расчёты и в затребованном пользователем виде предоставить ему результаты поиска. Если предположить, что таких запросов в единицу времени может поступать достаточно много, и сложность обработки каждого запроса велика, то для обеспечения приемлемого времени ответа необходимо иметь очень мощный, в смысле производительный, сервер. Во многих случаях один процессор может не справиться с такой работой, поэтому для увеличения производительности применяют многоядерные микропроцессоры и многопроцессорные системы.

Большое дисковое пространство

Современные жёсткие диски уже перешагнули рубеж ёмкости в 2 Тбайта, но при этом и минимальная ёмкость увеличилась до 500 Гбайт. Поэтому на рабочих станциях устанавливаются жёсткие диски достаточно большого объёма – от 500 Гбайт. Файловый сервер естественно должен иметь объём дискового пространства несколько больше, чем у рабочей станции, поэтому в сервер устанавливают несколько жёстких дисков. Для работы с большим количеством жёстких дисков все сетевые операционные системы поддерживают логическую единицу под названием том. Том представляет собой виртуальный жёсткий диск большого объёма, состоящий из нескольких жёстких дисков. Обращение к тому происходит как к обычному жёсткому диску, только большего размера. Это позволяет сетевым операционным системам управлять огромными дисковыми пространствами, например, ОС NetWare 4.1 допускает установку до 1024 жёстких дисков на сервер, которые можно объединить в 64 тома ёмкостью до 32 Тбайт, при этом общий объём доступного дискового пространства достигает 2048 Тбайт.

Достаточный объём ОЗУ

Поскольку сетевые операционные системы обрабатывают большое количество запросов, поддерживают гигантские объёмы дискового пространства, а при установке на сервер приложений – выполняют большое количество вычислений, то для обеспечения высокого быстродействия они должны некоторый объём информации хранить в оперативном запоминающем устройстве (ОЗУ). При этом, чем больше информации храниться в ОЗУ, тем быстрее будет работать сервер. Современные микропроцессоры типа

Xeon Core 2 Quad или Xeon i7 имеют возможность непосредственного обращения к памяти объёмом до 1 Тбайт, а специализированные серверные микропроцессоры имеют возможность адресации до 4096 Тбайт ОЗУ. Многие производители серверных платформ предлагают аппаратные возможности для установки этих объёмом памяти. Однако объём устанавливаемой памяти в сервер полностью зависит от финансовых возможностей приобретателя сервера, поскольку оперативная память всегда была и остаётся самым дорогим ресурсом вычислительной машины.

Высокая надёжность

Поскольку сервер является машиной коллективного пользования, то он должен обладать повышенной надёжностью. Ведь если выйдет из строя рабочая станция, то пострадает только один пользователь – владелец этой рабочей станции, а если откажет сервер, то может пострадать большое количество пользователей, поэтому на надёжность сервера следует обращать особое внимание. Надёжность сервера складывается из многих составляющих.

Во-первых, надёжность сервера напрямую зависит от надёжности компонентов его составляющих. Поэтому при покупке сервера следует обращать внимание на самые мелкие детали. Все компоненты должны быть известных и хорошо зарекомендовавших себя на рынке производителей, дающих хорошие гарантии. Следует избегать покупки отдельных компонент у мало или почти неизвестных производителей.

Во-вторых, единственным типом памяти, способным удовлетворить всё возрастающие и уже ставшие поистине гигантскими потребности современных компьютеров, является динамическая память DRAM – Dynamic Random Access Memory (динамическая память с произвольным доступом, произвольным потому, что возможно обращение к любой ячейке памяти по выбранному адресу). Постоянное совершенствование технологии изготовления памяти приводит к увеличению ёмкости кристаллов и снижению её стоимости.

В качестве элемента хранения информации в динамической памяти используется конденсатор, хранящий заряд (рис. 2.1). Хранение в элементе 1 или 0 определяется наличием или отсутствием заряда на конденсаторе. В современных микросхемах в качестве конденсатора используется p-n переход транзистора, поэтому такие микросхемы имеют очень высокую плотность упаковки.

Конденсатор из-за токов утечки со временем теряет свой заряд, поэтому элементы динамической памяти необходимо периодически подзаряжать. Конденсатор может случайно или утратить заряд, или приобрести его, при этом значение хранящего в нём бита станет ошибочным. Это может происходить

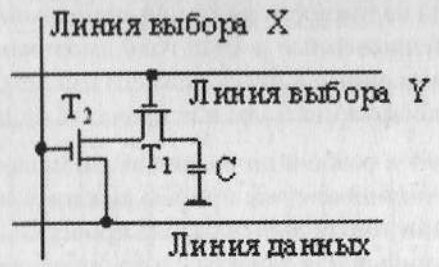


Рис. 2.1. ***Запоминающий элемент***

в случае выхода микросхемы из строя. Но даже в рабочей микросхеме конденсатор может изменить свой заряд. Обычно виновником таких ошибок является альфа-частица – ядро атома гелия (состоящего из двух протонов и двух нейтронов), выпущенная ядром тяжёлого атома в результате радиоактивного распада.

Альфа-частицы – отрицательный фактор, как правило, присущий устройствам памяти вычислительных машин, поскольку радиоактивные элементы в небольших количествах присутствуют почти во всех материалах. В частности, атомы урана и тория содержатся в пластмассовых корпусах обычных микросхем памяти. Таким образом, кристалл подвергается постоянной бомбардировке. Когда альфа-частица, обладающая достаточно высокой энергией, проникает в конденсатор, электроны покидают те места, которые они прежде занимали, и могут попасть на ближайшие конденсаторы. В результате конденсатор, в который попала альфа-частица, теряет свой заряд, а соседние конденсаторы могут, наоборот, его приобрести. В любом случае происходит ошибка.

Теоретические изыскания и практические исследования показали, что отдельно взятый конденсатор в микросхеме может подвергнуться бомбардировке с потерей заряда в среднем один раз в 100 миллионов лет, к счастью, достаточно редко. Однако память вычислительной машины может состоять из большого количества отдельных ячеек. Предположим, что среднее время до появления ошибки составляет ровно 100 миллионов лет. Тогда среднее время до появления ошибки в устройстве памяти ёмкостью всего 4 Гбайт будет равно 100 миллионов лет, поделённых на 34 359 738 368 (ровно столько бит соответствует 4 Гбайт), и составит всего один день. Не так уж и много и совсем неприемлемо для серверов. А экономичного способа защиты памяти от альфа-частиц нет.

Раз нельзя защититься от альфа-частиц, то, может быть, проще исправить ошибку. Для увеличения надёжности динамической памяти применяют корректирующие коды, предложенные в 1949 году американским учёным Ричардом Хэммингом и названные в честь него его именем, обладающие возможностью не только обнаруживать, но и исправлять одиночные ошибки.

Построение кода состоит в разбиении разрядов защищаемого слова на взаимно пересекающиеся подмножества, причём каждому подмножеству ставится в соответствие один контрольный разряд проверки на чётность. Количество контрольных разрядов для защиты слова определяется округлением в большую сторону до ближайшего целого числа результата, получаемого из следующего соотношения:

$$\log_2 N$$

где N общее количество разрядов в числе, включая информационные и контрольные разряды. Так, для защиты 128 разрядного слова, содержащего как информационные, так и контрольные биты, необходимо 7 контрольных разрядов. Контрольные разряды перемежаются с информационными и располагаются в позициях с номерами: $N_i = 2^i$ (i=0, 1, 2, ...), то есть в позициях с номерами 1, 2, 4, 8, 16, 32, 64 и т.д.

Формирование подмножеств производится на основе анализа номера разряда при записи его в двоичной форме. В первое подмножество включаются все разряды, имеющие в номере единицу в младшем разряде – xxxxx1, во второе – единицу в следующем разряде xxxx1x, в третье – единицу в третьем разряде – xxx1xx, и так далее. Затем в каждом подмножестве подсчитывается количество единиц и в соответствующий контрольный разряд записывается единица, если это число нечётно, и ноль – если чётно.

На рис. 2.2 приведён пример, как из 15-ти разрядного слова формируется 20-ти разрядный результат. Проверка слова на правильность сводится к вычислению синдрома ошибки $S=s_5s_4s_3s_2s_1$, по которому можно судить о наличии ошибки и её месте. Число разрядов в синдроме ошибки равно числу контрольных разрядов. В приведённом на рис. 2.2 примере синдром ошибки будет содержать пять разрядов. Для вычисления разряда s_i синдрома ошибки необходимо сложить по модулю 2 разряды, относящиеся к i-тому подмножеству, включая i-й контрольный разряд кода Хэмминга. Если все разряды кода синдрома ошибки равны нулю S = 0 0 0 0 0, то в закодированном слове ошибки нет. Если значение синдрома ошибки не равно нулю, то в закодированном слове имеется ошибка, а значение синдрома указывает на ошибочный разряд (им может быть и контрольный разряд). Для исправления ошибки достаточно проинвертировать значение указанного разряда.

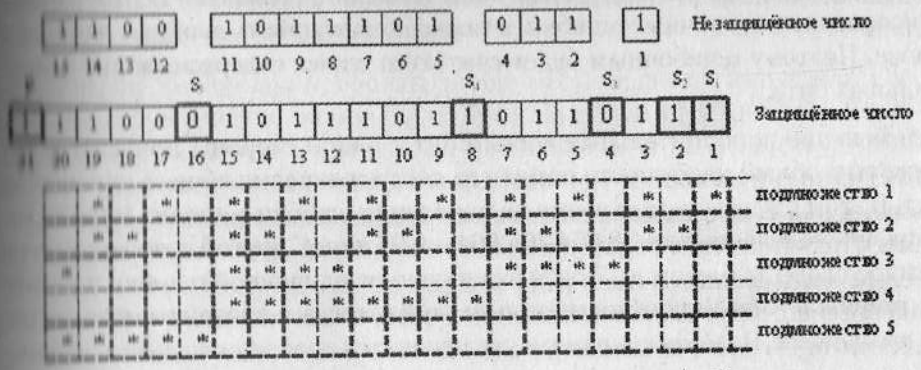


Рис. 2.2. ***Кодирование 15-ти разрядного слова по коду Хэмминга***

Если в закодированном слове будут искажены два и более разрядов, то синдром ошибки укажет на разряд, не содержащий ошибки. Для определения места одиночной и факта двойной ошибки вводят ещё один дополнительный разряд P для проверки на чётность всего закодированного слова (на рис. 2.1 это 21-й разряд). Если произойдёт одиночная ошибка, то её выявит проверка на чётность всего слова, а точное место ошибки укажет синдром ошибки.

При искажении содержимого двух разрядов общая проверка на чётность ошибку не зафиксирует, а синдром ошибки будет отличен от нуля, это говорит о том, что ошибка есть, но её точное место определить невозможно. Ошибки кратности 3,5,7,... будут восприниматься как одиночные ошибки, однако вероятность их появления на несколько порядков меньше одиночной.

При проверке слова система декодирования принимает следующие решения в зависимости от значения общей проверки и синдрома ошибки:

Общая чётность P	Синдром ошибки S	Решение
0	0	Ошибок нет
0	S	Неисправимая ошибка кратности 2, 4, 6,...
P	0	Неисправимая пакетная ошибка в 2'-1 младших разрядах
P	S	Ошибка кратности 1, 3, 5,... (исправление одиночной)

Для контроля ошибок можно использовать и другие более сложные коды, например Рида-Соломона.

Так как в современных микропроцессорах используются слова размером в 64 бита, то для их защиты с использованием корректирующих кодов Хэмминга необходимо ещё семь бит. Поскольку микросхемы, применяемые в модулях памяти, имеют размер слова 8 бит, то восьмой разряд используется для

специальных нужд. Рассмотренные коды Хэмминга позволяют обнаруживать и исправлять одиночные ошибки, а также обнаруживать двойные ошибки в слове. Поэтому ошибочным будет считаться слово, содержащее два неправильных бита.

Применение дополнительных восьми бит с одной стороны приведёт к понижению общей надёжности памяти за счёт увеличения общего количества бит. В этом случае среднее время до появления ошибки составит 100 миллионов лет, поделённых на 38 654 705 664 – 0,94 дня. С другой стороны, теперь каждое слово размером в 72 бита защищено от одиночной ошибки в любом из разрядов. Сбой происходит только тогда, когда в слове возникают сразу две ошибки. Надёжность памяти увеличивается пропорционально величине:

$$\sqrt{\frac{\pi}{2}N}$$

где N количество слов в памяти. При объёме памяти в 4 Гбайт и длине слова 64-бит общее количество слов будет равно 536 870 912. Тогда с использованием кодов Хэмминга надёжность памяти возрастёт в 29 032 раза и составит около 74,7 года. Это уже вполне приемлемая величина. Во всяком случае, за гарантийное время эксплуатации машины, около 10 лет, память ни разу не «сбойнёт».

Сам модуль, имеющий дополнительную микросхему, называется ECC (система исправления ошибок), но он только может хранить дополнительную информацию, для исправления же ошибок необходимо специальное устройство, вычисляющее коды Хэмминга, которое встраивается в контроллер динамической памяти. Но производители модулей ECC прекрасно понимают, для каких целей потребитель покупает такие модули – для установки в сервер и зарабатывания денег путём предоставления услуг клиентам. Поэтому они считают, что покупатель обязан поделиться с ними своей прибылью, и увеличивают цены на такие модули в 2 – 3 раза.

В-третьих, жёсткие диски являются самым ненадёжным компонентом системы, потому что имеют в своём составе движущиеся элементы - двигатель, пластины, коромысло. По мере работы жёсткого диска на подшипники всех движущиеся частей воздействуют значительные силы, приводящие к износу последних, а это может привести к выходу жёсткого диска из строя. Вообще-то, ведя разговор о жёстком диске, следует иметь ввиду, что он обязательно выйдет из строя, и как показывает практика, в самый что ни наесть неподходящий момент – вы только что закончили большой труд, например курсовой проект, написали диплом, а на следующий день жёсткий диск не подаёт никаких признаков жизни, вашему разочарованию нет предела. Поэтому вопрос

о надёжности жёсткого диска необходимо ставить так – не диск может выйти из строя, а когда он выйдет из строя.

Для серверов все компании, производящие жёсткие диски, выпускают специальные серии дисков, обладающих очень высокой надёжностью, имеющих параметр наработки на отказ до 100 000 часов, что соответствует примерно 11 годам непрерывной работы. Однако при установке в сервер 1024 дисков общая надёжность дисковой системы снижается ровно во столько раз, сколько установлено дисков, и в нашем примере будет составлять всего около 4-х дней. Получается, что каждые 4 дня один из дисков будет выходить из строя – такая ситуация совершенно не приемлема даже для рабочей станции, не говоря уже о сервере. Даже если наработка на отказ одного диска будет составлять 1 000 000 часов, то этого всё равно недостаточно, поскольку каждые 39 дней один из дисков будет выходить из строя.

Для повышения надёжности дисковых систем применяют метод, аналогичный заполнению оперативной памяти, только он называется «избыточный массив жёстких дисков» – RAID. Существует несколько уровней RAID. Изначально RAID был задуман для создания на базе нескольких жёстких дисков диска большего объёма с увеличенной скоростью доступа. Рассмотрим дисковые массивы всех уровней.

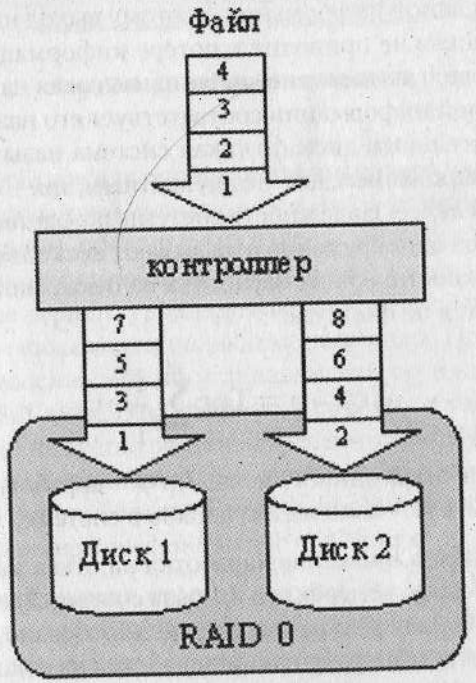


Рис. 2.3. ***Дисковый массив RAID 0***

RAID 0. Дисковый массив нулевого уровня представляет собой несколько параллельно работающих дисков, при этом файл, подлежащий записи, делится на части (блоки). Блоки записываются на все диски, причём на каждый диск записываются блоки кратные номеру диска. Так, если в массиве установлены 3 диска, то на первый диск записываются блоки с номерами 1, 4, 7, 10 ..., на второй диск записываются блоки с номерами 2, 5, 8, 11 ..., а на третий диск блоки с номерами 3, 6, 9, 12 ... Запись на все диски осуществляется параллельно. На рис. 2.3 приведён пример массива, содержащего два диска. Достоинством данного массива является высокая скорость обмена информацией, пропорциональная количеству установленных дисков, и большой объём дискового пространства, равный суммарной ёмкости дисков. К недостаткам следует отнести низкую надёжность, поскольку выход одного диска из строя приводит к полной потере всей информации. Однако для тех применений, в которых определяющими являются скорость и объём, применение таких массивов оправдано.

RAID 1. Дисковый массив первого уровня предназначен для повышения надёжности жёсткого диска и представляет собой несколько параллельно работающих дисков, при этом файл, подлежащий записи, записывается сразу на несколько дисков – дублируется (рис. 2.4). В этом случае создается несколько копий записанной информации, поэтому выход из строя одного или даже нескольких дисков не приводит к потере информации. Достоинством массива первого уровня, несомненно, является высокая надёжность, количество копий записанной информации соответствует его названию. Если в системе один дополнительный диск, то такая система называется системой с однократным резервированием, два – с двукратным, три – с трёхкратным резервированием и так далее. Надёжность системы возрастает благодаря тому, что два и более дисков одновременно отказывают несколько реже, и как следует из теории надёжности, общая наработка на отказ системы с резервированием рассчитывается по формуле:

$$Tср - р = Tрс \sum_{i=1}^{m} \frac{1}{i}$$

где: *Tср* – наработка на отказ одного диска, *Tср-р* – наработка на отказ массива с резервированием, а *m* – количество дисков в системе.

При наличии двух дисков в массиве наработка на отказ массива возрастает в 1,5 раза, трёх – в 1,8 раза, четырёх – в 2,1 раза соответственно. Таким образом, резервирование более трёхкратного не целесообразно, поскольку стоимость системы возрастает многократно, а наработка на отказ незначительно. При этом общая ёмкость массива равна ёмкости одного диска. Для систем с резервированием необходимо использовать одинаковые диски.

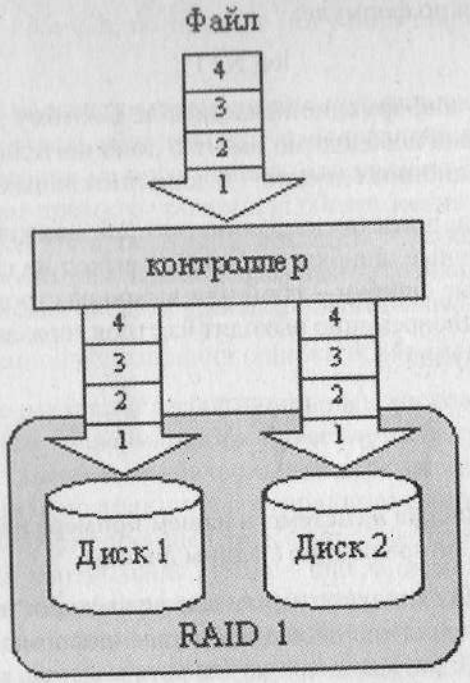


Рис. 2.4. ***Дисковый массив RAID 1***

В зарубежной литературе для определения дискового массива первого уровня, наряду с дублированием, используется ещё и термин «зеркалирование», по аналогии с зеркалом, поскольку на втором диске создается «зеркальная» копия информации. Это синонимы одного и того же процесса. Узким местом в дисковом массиве первого уровня является контроллер, при выходе его из строя весь массив оказывается не работоспособным. Поэтому для увеличения надёжности массива можно устанавливать по индивидуальному контроллеру на каждый диск. Это, конечно, приводит к увеличению надёжности массива, однако не всегда рационально, потому что электронные компоненты по сравнению с механическими устройствами имеют наработку на отказ минимум на порядок больше, поэтому нет надобности дублировать высоконадёжный компонент на фоне менее надёжных.

RAID 2. Дисковый массив второго уровня представляет собой многодисковый массив с системой обнаружения и исправления ошибок на основе кодов Хэмминга. Как и в случае с оперативной памятью, описанной выше, количество дополнительных дисков, необходимых для защиты информаци-

онных, определяется по формуле:

$$\log_2 N+1$$

где: N – количество информационных дисков. Поэтому для защиты 16 информационных дисков необходимо иметь 5 дополнительных дисков, а для защиты 32 информационных дисков – 6 дополнительных соответственно.

Система из 21 диска, представленная на рис. 2.5, позволяет обнаруживать и исправлять одиночные ошибки – сбой или выход из строя одного диска и определять двойные ошибки – сбой или выход из строя двух дисков. Поскольку два диска одновременно выходят из строя гораздо реже одного, то в соответствии с формулой

$$\sqrt{\frac{\pi}{2}N}$$

где N – количество дисков в системе, в нашем примере надёжность массива возрастает в 5,7 раза, по сравнению с одним диском.

К достоинствам RAID 2 следует отнести высокую скорость передачи информации, вследствие параллельной записи на все информационные диски (в нашем примере – в 16 раз), и высокую ёмкость массива, которая равняется сумме всех 16 информационных дисков, а также высокую надёжность. Реализация дискового массива RAID 2 предусматривала наличие специального весьма дорогостоящего контроллера, поскольку запись на диски должна

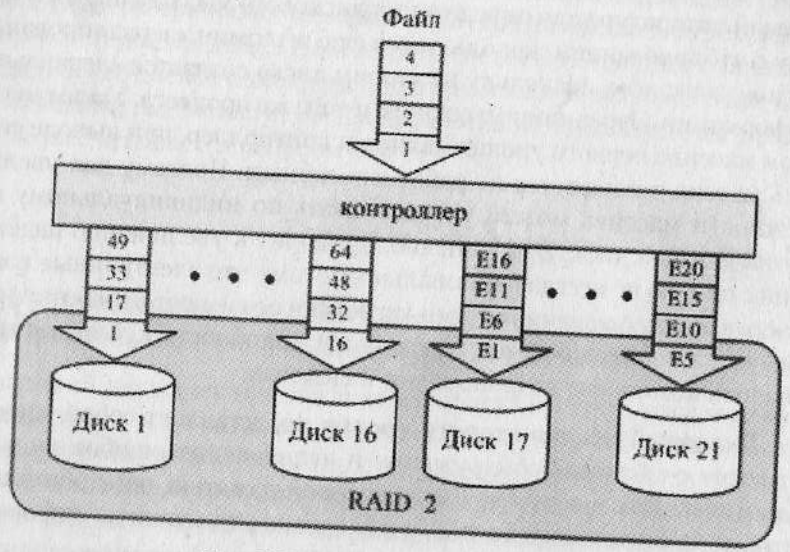


Рис. 2.5. ***Дисковый массив RAID 2***

быть производиться побитно, которые не получили широкого распространения.

RAID 3. Дисковый массив третьего уровня представляет собой многодисковый массив с системой обнаружения и исправления ошибок. В отличие от дискового массива второго уровня, использующего коды Хэмминга, в дисковом массиве третьего уровня для обнаружения и восстановления информации применяется простейший механизм – поразрядное сложение (сложение без переноса) или, как ещё называют эту операцию, – исключающее ИЛИ. Контрольная информация располагается на отдельном диске.

Для пояснения принципа исправления ошибки обратимся к рис. 2.6.

Допустим, в системе имеется 8 информационных дисков и один контрольный. Вся поступающая информация разбивается на байты и записывается на информационные диски, а на контрольный диск записывается контрольная сумма, вычисленная по принципу поразрядного сложения. В примере на рис. 2.6 на каждый информационный диск записывается информация от 000 до 111. Вычисленная контрольная сумма – 000 записывается на контрольный диск. Допустим, что информационный диск под номером 3 вышел из строя. Тогда при чтении над содержимым всех информационных дисков и контрольного диска производится операция поразрядного сложения, в результате получается значение 011, что соответствует утраченной информации. Таким образом, удаётся восстановить информацию.

В дисковом массиве третьего уровня, представленном на рис. 2.7, вся поступающая информация записывается на информационные диски блоками, а на контрольный – контрольная сумма.

Диски	Информация	
	Запись	Чтение
0	000	000
1	001	001
2	010	010
3	011	xxx
4	100	100
5	101	101
6	110	110
7	111	111
K	000	000

Рис. 2.6. ***Вычисление контрольной суммы и исправление ошибки***

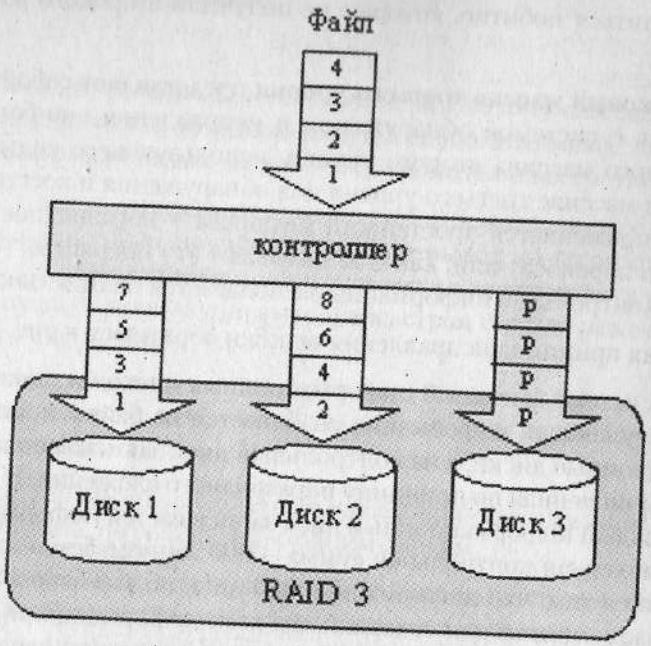


Рис. 2.7. ***Дисковый массив RAID 3***

Блоки данных, записываемые на диск, имеют малый размер, равный принятому в данной операционной системе кластеру, поделённому на количество информационных дисков массива. Самое большое достоинство метода вычисления контрольной суммы по модулю два заключается в том, что результат всегда равен размеру записываемого блока данных, поэтому, независимо от количества информационных дисков, для хранения контрольных сумм всегда достаточно одного диска. На практике количество информационных дисков редко превышает 16.

К достоинствам дискового массива третьего уровня следует отнести высокую скорость обмена информацией, пропорциональную количеству информационных дисков. Простой механизм вычисления контрольной суммы и восстановления потерянной информации. Большую ёмкость массива, равную сумме всех информационных дисков. Высокую надёжность при небольшой избыточности, поскольку потеря информации происходит только при отказе двух и более дисков одновременно. Поэтому дисковый массив, содержащий всего три диска, имеет наработку на отказ в 2 раза больше, по сравнению с одним диском.

В связи с тем, что размер блока невелик, то при каждой операции ввода/вывода происходит обращение ко всем дискам массива, что не позволяет производить несколько операций одновременно. При работе с короткими записями из-за невозможности синхронизации вращения дисков на разных приводах неизбежно уменьшение скорости обмена. Поэтому дисковый массив третьего уровня хорошо подходит для однозадачных операционных систем, работающих с большими файлами.

RAID 4. Дисковый массив четвёртого уровня представляет собой массив третьего уровня, в котором информация на информационные диски записывается блоками большого размера, соизмеримого с размером кластера. Это обстоятельство позволило увеличить скорость чтения при случайных обращениях к файлам большого размера вследствие того, что файл занимает меньше блоков – в идеале один, а это в свою очередь позволяет производить несколько операций чтения одновременно. Однако при записи контрольная информация должна каждый раз обновляться на контрольном диске, поэтому одновременное выполнение операций записи невозможно. Поскольку проблема одновременного выполнения нескольких записей не решена, то дисковые массивы четвёртого уровня не нашли широкого применения.

RAID 5. Дисковый массив пятого уровня представляет собой многодисковый массив с системой обнаружения и исправления ошибок, принятой в дисковом массиве третьего уровня. В отличие от дискового массива третьего уровня, использующего один диск для хранения контрольных сумм, в дисковом массиве пятого уровня контрольная информация равномерно распределяется по всем дискам (рис. 2.8).

Кроме этого, дисковый массив пятого уровня использует блоки большого размера, как в дисковом массиве четвёртого уровня. В результате этих нововведений удалось решить проблему выполнения нескольких операций одновременно. Правда, такой подход не всегда гарантирует возможность выполнения нескольких операций записи одновременно. Это связано с возможностью попадания операций записи контрольных сумм на один диск. Поэтому для уменьшения таких ситуаций целесообразно применение большого количества дисков. Разработчики рекомендуют в дисковых массивах пятого уровня применять не менее пяти дисков.

RAID 6. Дисковый массив шестого уровня представляет собой многодисковый массив с системой обнаружения и исправления ошибок, как в дисковом массиве третьего уровня. Но для увеличения надёжности производится вычисление двух контрольных сумм P и Q (рис. 2.9) на основе разных алгоритмов. Первая контрольная сумма P вычисляется по значениям данных записываемых блоков в массив на основе операции "сумма по модулю два", а вторая контрольная сумма Q вычисляется по значениям тех же записывае-

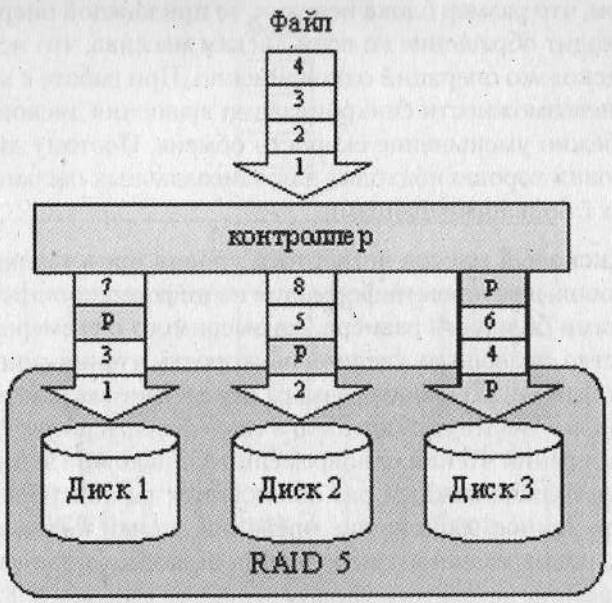


Рис. 2.8. ***Дисковый массив RAID 5***

мых блоков в массив, что и первая контрольная сумма, но на основе другого алгоритма. После вычисления контрольных сумм они обе записываются на рядом расположенные жёсткие диски. В результате алгоритм восстановления P+Q позволяет восстановить информацию при одновременном отказе двух любых дисков массива. Потеря же информации возможна только в случае отказа трёх и более дисков одновременно.

Применение дискового массива RAID 6 оправдано только в тех случаях, когда необходимо обеспечить очень высокую надёжность хранения информации.

Ранее были рассмотрены базовые дисковые массивы, для повышения надёжности иногда применяются составные массивы. Это и понятно, ведь ни один из рассмотренных выше массивов не обладает идеальными характеристиками. Чаще всего используются сочетания быстрого массива RAID 0 с надёжными – RAID 1, 3 или 5. Итоговый массив действительно обладает улучшенными характеристиками, но за всё приходится платить повышением стоимости и сложности решения.

Составной массив строится из нескольких массивов, объединённых в один более сложный массив. Маркируются такие массивы двумя цифрами XY. Запись типа X+Y или XY означает, что сначала диски объединяются в массив уровня X, а затем несколько массивов RAID X объединяются в массив уровня Y.

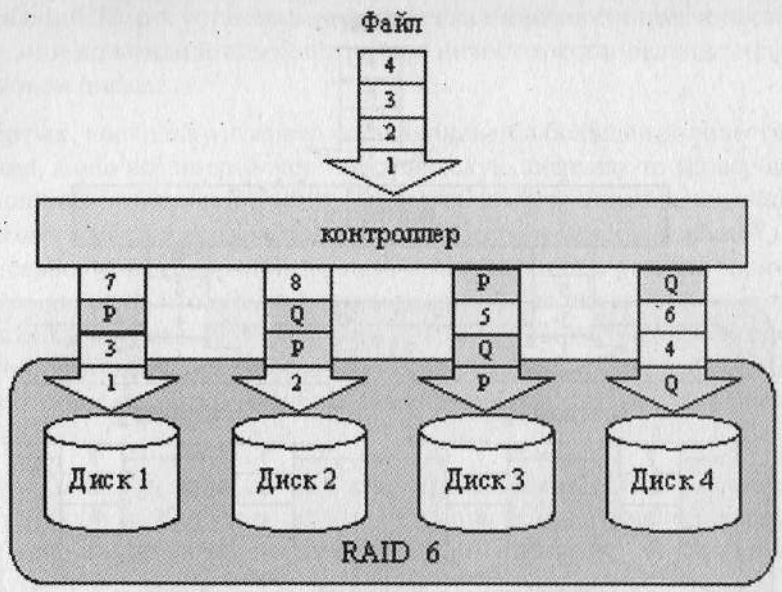


Рис. 2.9. ***Дисковый массив RAID 6***

RAID 1+0 (10). Данный составной массив состоит из нескольких дисковых массивов первого уровня, объединённых в дисковый массив нулевого уровня (рис. 2.9). Возможно создание дискового массива RAID 01. Принципиально эти два составных массива очень схожи, имеют одинаковые характеристики и явных преимуществ друг перед другом не имеют. Поэтому каждый потребитель самостоятельно решает, какой тип составного массива ему больше подходит. В любом случае для организации составного массива потребуется несколько контроллеров. Правда, в последнее время многие производители материнских плат предлагают недорогие программные решения дисковых массивов, что совершенно не подходит для серверов из-за высокой вычислительной нагрузки на процессор.

Поскольку контроллеры RAID-массивов должны производить большое количество вычислений в темпе поступления информации, то при создании контроллеров используют микропроцессоры младших моделей, в связи с тем, что они имеют достаточное быстродействие.

RAID 50. Такой массив обладает крайне высокой производительностью, поскольку запись или чтение производится при обращении одновременно ко всем дискам (рис. 2.10). Высокая надёжность массива RAID 50 определяется тем, что каждый из дисковых массивов RAID 5 обладает повышенной надёжностью. Однако при выходе из строя любого массива RAID 5 вся система становится неработоспособной.

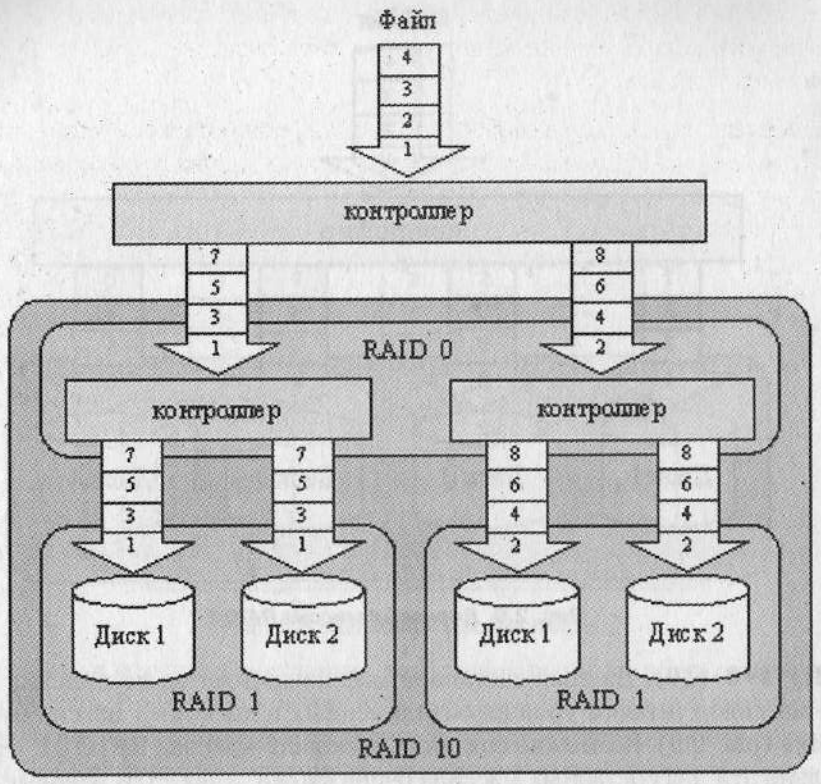


Рис. 2.10. ***Дисковый массив RAID 10***

Принципиально нет никаких ограничений в построении составных массивов. Можно реализовать составной массив и более высоких уровней, например RAID 501. В этом случае несколько дисковых массивов пятого уровня объединяются в несколько массивов нулевого уровня, которые в свою очередь объединяются в дисковый массив первого уровня. Несомненно, что полученный массив будет обладать лучшими характеристиками по производительности и надёжности, чем любой массив, из которых он составлен. Но нужен ли такой массив, если он будет неимоверной цены, вот в чём вопрос. Поэтому для серверов каждый потребитель должен исходя из сметы подобрать подходящий массив, который бы устраивал его и по надёжности, и по производительности.

Для дальнейшего повышения надёжности дисковых массивов все контроллеры RAID 3 и RAID 5 поддерживают так называемый режим горячей замены. В этом режиме система допускает без выключения (остановки работы массива) массива извлечь отказавший диск из массива и установить на его

место рабочий. После установки нового диска в массив система или самостоятельно, или по команде администратора начнёт восстановление информации на новом диске.

В-четвёртых, поскольку в сервер устанавливается большое количество оборудования, а оно всё потребляет электрическую энергию, то сервер должен иметь мощный источник питания. Как и любое оборудование, источник питания может выйти из строя, поэтому для обеспечения надлежащей надёжности в сервер устанавливают два источника питания в режиме горячей замены. Это означает, что оба источника находятся во включённом состоянии и работают по очереди, например часа по два-три. Такой режим обеспечивает равномерный износ источников питания. А режим горячей замены позволяет без остановки сервера заменить вышедший из строя источник на новый рабочий.

Электрические сети сами по себе для сервера являются повышенным источником проблем. Качество питающего напряжения оставляет желать лучшего. Декларируемые значения 220 В и 50 Гц никогда не выполняются,

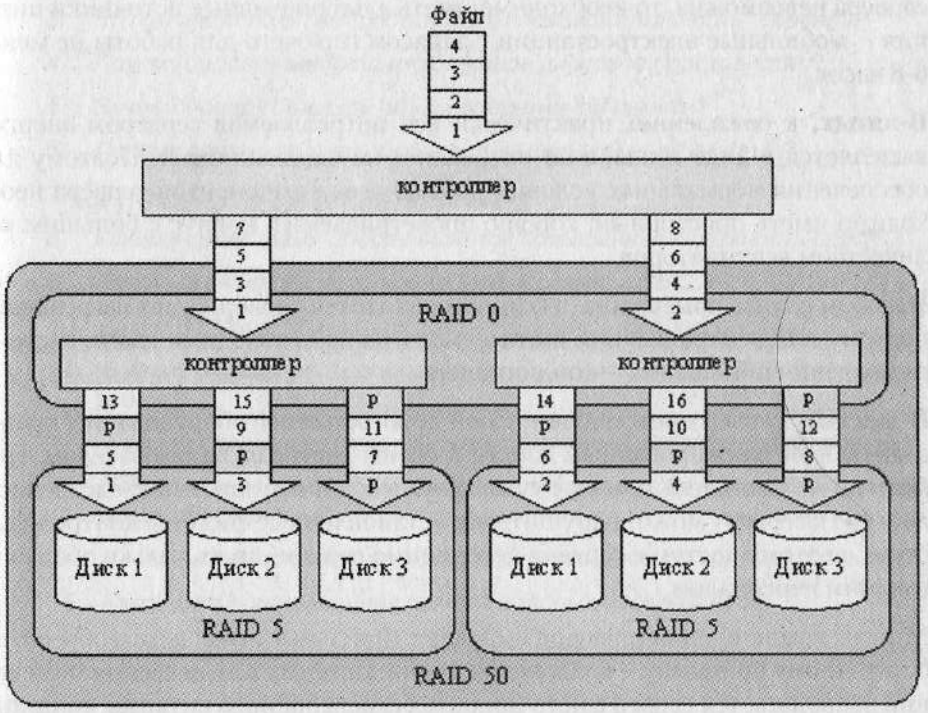


Рис. 2.11. ***Дисковый массив RAID 50***

кроме этого, электрические компании частенько выключают подачу электроэнергии без предупреждения. Естественно, что для сервера внезапное пропадание питающего напряжения равносильно катастрофе, в связи с тем, что прерываются все вычислительные процессы, а самое главное, что могут прерваться процессы записи на жёсткие диски. Это может привести к потере не только информации, но и разрушению жёстких дисков.

Из всего выше сказанного следует, что для нормальной работы сервера необходимо иметь бесперебойные источники питания, представляющие собой набор аккумуляторов со схемами преобразования. Источник бесперебойного питания выполняет ещё и функцию обеспечения питающего напряжения в заданных пределах. Ёмкость аккумуляторных батарей должна быть достаточна для того, чтобы в случае пропадания питания в электрической сети нормально завершить все процессы, сообщить рабочим станциям о том, что сервер выключается, и корректно остановиться, обычно на это требуется 10-15 минут.

При любых авариях электроснабжающие компании гарантируют восстановление энергоснабжения в течение четырёх часов. Поэтому если остановка сервера невозможна, то необходимо иметь альтернативные источники питания – мобильные электростанции с запасом горючего для работы не менее 6-8 часов.

В-пятых, к сожалению, практически вся потребляемая сервером энергия выделяется в виде тепла, а он потребляет не один киловатт. Поэтому для обеспечения нормальных условий работы всех компонентов сервера необходимо иметь просторный, хорошо проветриваемый корпус с большим количеством вентиляторов.

В любом случае помещение, в котором находится сервер, будет нагреваться, поэтому для нейтрализации выделяемого тепла необходимо иметь систему охлаждения помещения – кондиционер.

В-шестых, сервер представляет собой дорогостоящее оборудование, предоставляющее ранжированный доступ к своим ресурсам пользователям. Грамотный пользователь или ещё хуже злоумышленник при непосредственном доступе к серверу может нарушить все установленные системы контроля доступа, поэтому доступ к серверу необходимо ограничивать только обслуживающим персоналом.

Многие компании для серверов выделяют отдельные помещения, обычно по остаточному принципу – кладовки без окон. Поэтому в этом тесном помещении располагается сервер с источниками бесперебойного питания, кондиционером и всяким дополнительным хламом для сервера. Ключ от помещения обычно находится в ведении системного администратора.

В-седьмых, всё вышеизложенное справедливо при наличии достаточного количества финансов. Однако именно здесь будущего администратора и подстерегают огромные проблемы. Ни один себя уважающий директор компании никогда не даст столько денег, поскольку он считает, что раз сервер работает, значит всё нормально и нечего тратить такие огромные деньги на «кусок железа». Поэтому инстинкт самосохранения должен подсказать администратору необходимость вести с директором «разъяснительную» работу и убеждать его в необходимости вкладывать деньги в сервер. В противном случае при поломке сервера останется крайним администратор, и все шишки свалятся на него. А директор будет утверждать, что виноват именно администратор, раз не смог его убедить в необходимости вложения средств в сервер.

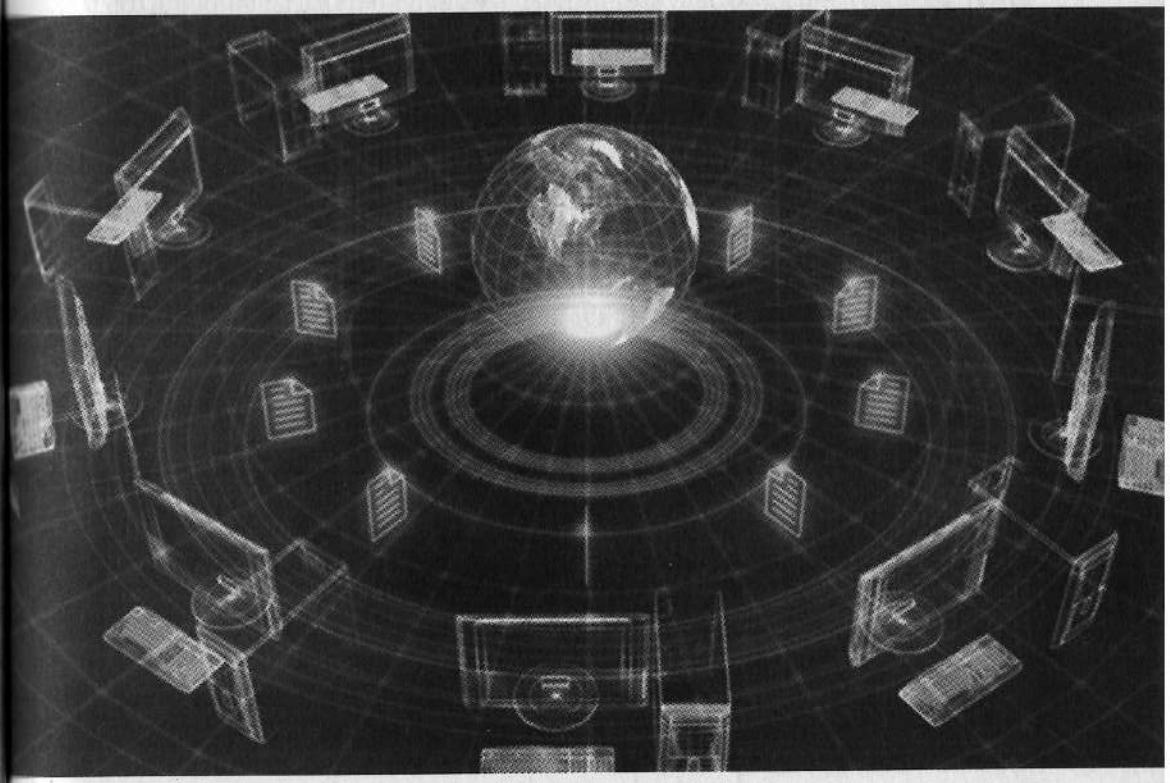
Контрольные вопросы

1. *Назовите основные компоненты сети.*
2. *Какие требования предъявляются к рабочим станциям со стороны сети?*
3. *Что означает понятие "достаточная производительность" сервера?*
4. *Почему серверу необходимо большое дисковое пространство?*
5. *Зачем серверу большой объём оперативной памяти?*
6. *Почему сервер должен обладать повышенной надёжностью?*
7. *От чего зависит надёжность сервера?*
8. *Какими средствами обеспечивается повышенная надёжность ОЗУ?*
9. *Почему в серверах используют модули памяти с ECC?*
10. *Каким образом можно повысить надёжность жёстких дисков?*
11. *В каких случаях целесообразно использовать дисковый массив нулевого уровня?*
12. *Какими преимуществами обладает дисковый массив первого уровня?*
13. *Почему в последнее время отказались от использования дисковых массивов второго уровня?*
14. *В чём заключается достоинство дискового массива третьего уровня?*
15. *Какой метод исправления ошибок используется в RAID 3?*
16. *Почему дисковый массив третьего уровня целесообразно использовать в однозадачных операционных системах?*
17. *По какой причине дисковые массивы четвёртого уровня не получили широкого распространения?*

18. *Назовите основное отличие дискового массива пятого уровня от дискового массива третьего уровня?*
19. *Почему дисковый массив пятого уровня хорошо подходит для использования с многозадачными операционными системами?*
20. *За счёт чего дисковый массив шестого уровня обладает высочайшей надёжностью?*
21. *В чём заключается отличие дискового массива 10 от массива 01?*
22. *Как устроен дисковый массив 50?*
23. *Какие преимущества даёт использование дискового массива 50?*
24. *Каким образом можно уменьшить негативные последствия от аварии в сети энергоснабжения сервера?*
25. *Зачем в серверной устанавливают кондиционер?*

Глава 3.

КОММУНИКАЦИОННОЕ ОБОРУДОВАНИЕ



Коммуникационное оборудование сетей является одной из главных составляющих сетей передачи данных. Именно от него зависит скорость, надёжность и стоимость передачи информации между узлами сети. Поэтому изучение коммуникационного оборудования является основной задачей освоения сетей. Большинство дальнейшего материала настоящей книги посвящено именно этой проблеме. Коммуникационное оборудование сетей передачи данных состоит из двух основных компонент: линий связи и оборудования, обеспечивающего передачу данных по этим линиям связи.

3.1. Линии связи

Линия связи состоит в общем случае из физической среды, по которой передаются электрические информационные сигналы. Независимо от передающей среды, линия связи влияет на передаваемые по ней сигналы. Для рассмотрения влияния линии на сигнал обратимся к эквивалентной схеме линии, представленной на рис. 3.1.

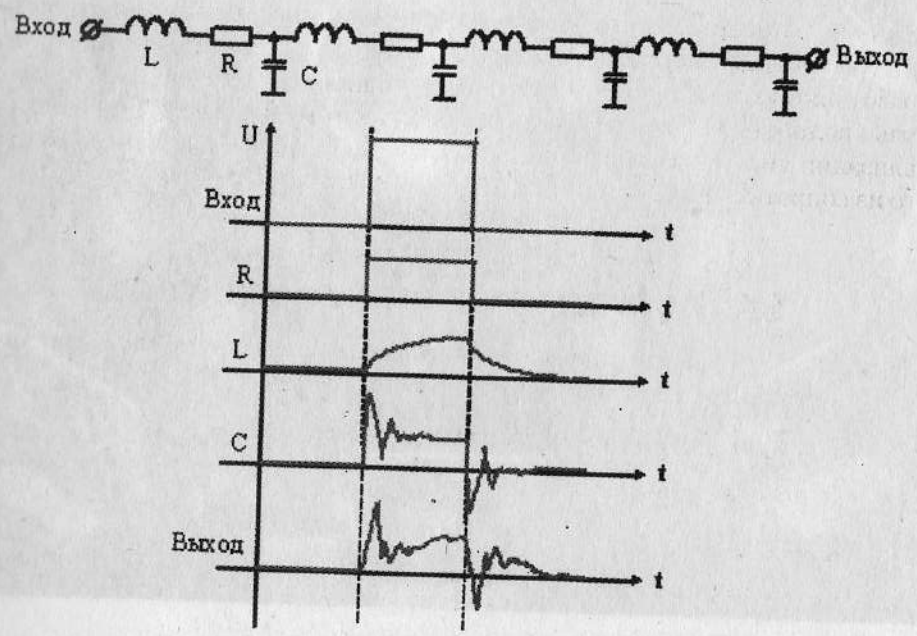


Рис. 3.1. ***Влияние природы сопротивления искажения формы импульса***

Любую линию можно представить в виде последовательно соединённых распределённых основных сопротивлений: активного R, индуктивного L и ёмкостного C. Каждое из них вносит свою посильную лепту в искажение формы входного сигнала. На рис. 3.1 показано влияние каждого из перечисленных сопротивлений на прямоугольный входной сигнал и их суммарное влияние.

Если бы линия состояла их одного активного сопротивления R, то за счёт выделения тепла на нём амплитуда входного сигнала на выходе уменьшилась.

Индуктивное сопротивление L представляет собой интегрирующее звено, которое сначала накапливает энергию, а потом её отдаёт, поэтому если бы линия состояла из одного индуктивного сопротивления, то входной сигнал на выходе принял бы форму далёкую от прямоугольного.

Ёмкостное сопротивление C представляет собой дифференцирующее звено, и его влияние на входной прямоугольный импульс особенно сказывается на фронтах в виде всплесков напряжения разного знака.

Поскольку в линии присутствуют все три сопротивления, то входной сигнал ослабляется и искажается до неузнаваемости. Поэтому приёмное устройство должно иметь возможность выделить такой искажённый сигнал на фоне естественных помех линии. Собственно, присущими линии искажениями и определяется максимальная дальность и скорость передачи. Разработчики пытаются свести к минимуму влияние этих сопротивлений, однако полностью их исключить не удаётся, поэтому в каждой конкретной реализации линии, разработчики стараются уменьшить влияние только одного из сопротивлений.

3.2. Характеристики линий связи

Качество линии связи определяется целым рядом параметров. К основным характеристикам линий связи относятся:

- амплитудно-частотная характеристика;
- полоса пропускания;
- затухание;
- помехоустойчивость;
- перекрестные наводки на ближнем конце линии;
- пропускная способность;

- достоверность передачи данных;
- удельная стоимость.

В первую очередь разработчика вычислительной сети интересуют пропускная способность и достоверность передачи данных, поскольку эти характеристики напрямую влияют на производительность и надёжность создаваемой сети. Пропускная способность и достоверность – это характеристики как линии связи, так и способа передачи данных. Поэтому если способ передачи (протокол) уже определён, то эти характеристики известны.

Для определения характеристик линии связи часто используют анализ её реакций на некоторые эталонные воздействия. Такой подход позволяет достаточно просто определять характеристики линий связи любой природы, не прибегая к сложным теоретическим исследованиям. Чаще всего в качестве эталонных сигналов для исследования реакций линий связи используются синусоидальные сигналы различных частот. Это связано с тем, что синусоидальные сигналы часто встречаются в технике.

Из теории сигналов следует, что любой периодический процесс можно представить в виде суммы синусоидальных колебаний различных частот и различных амплитуд. Каждая составляющая синусоида называется гармоникой, а набор всех гармоник называют спектром сигнала.

Для сигналов, которые хорошо описываются аналитически, спектр легко вычисляется на основе формул Фурье. Для сигналов произвольной формы, встречающихся на практике, спектр можно найти с помощью специальных приборов – спектральных анализаторов, которые измеряют спектр реального сигнала и отображают амплитуды составляющих гармоник.

Амплитудно-частотная характеристика линии (рис. 3.2) показывает, как затухает амплитуда синусоиды на выходе линии связи по сравнению с амплитудой на её входе во всём диапазоне частот. Вместо амплитуды в этой характеристике часто используют такой параметр сигнала, как его мощность.

Полоса пропускания определяется на основе амплитудно-частотной характеристики, путём проведения линии параллельно оси частоты на уровне 0,5 максимальной величины амплитудной характеристики. В точках пересечения с амплитудно-частотной характеристикой проводятся вертикально линии до пересечения с осью частот. Полученный таким образом диапазон частот и называется полосой пропускания линии связи. Величина полосы пропускания в наибольшей степени влияет на скорость передачи информации по линии связи.

Затухание определяется как относительное уменьшение амплитуды или мощности сигнала определенной частоты на выходе линии. Таким образом,

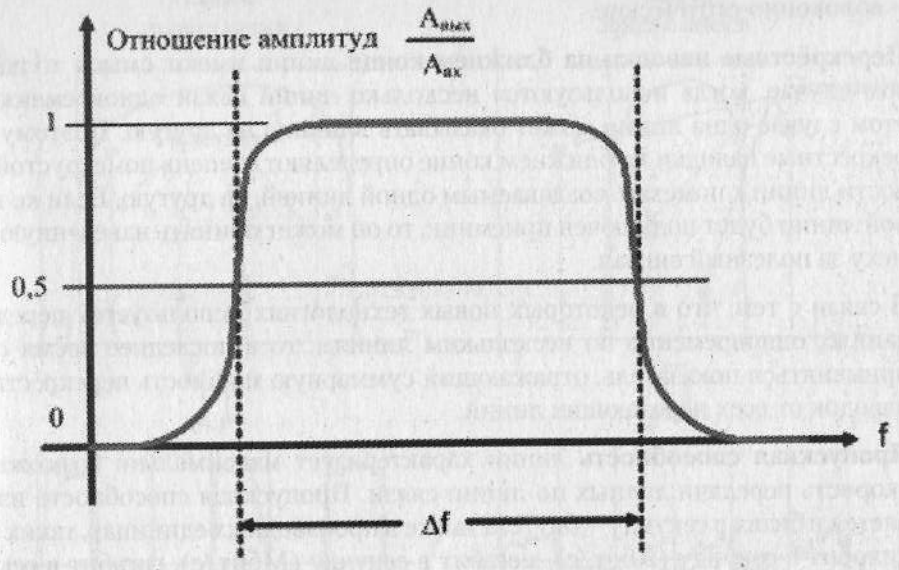


Рис. 3.2. ***Амплитудно-частотная характеристика***

затухание представляет собой одну точку в амплитудно-частотной характеристике линии. Поскольку заранее известна основная частота передаваемого сигнала, то достаточно знать затухание на этой частоте, чтобы приблизительно оценить искажения передаваемых по линии сигналов. Более точные оценки возможны при знании затухания на нескольких частотах, соответствующих нескольким основным гармоникам передаваемого сигнала.

Затухание А измеряется в децибелах и вычисляется по следующей формуле:

$$A = 10\log_{10}\frac{P_{вых}}{P_{вх}}$$

где $P_{вых}$ – мощность сигнала на выходе линии, а $P_{вх}$ – мощность сигнала на входе линии соответственно. Поскольку мощность выходного сигнала всегда меньше, чем мощность входного сигнала (сказывается влияние активного сопротивления линии), затухание всегда является величиной отрицательной.

Помехоустойчивость линии определяет её способность сопротивляться внешним помехам. Помехоустойчивость линии зависит от типа используемой физической среды, а также от экранирующих и подавляющих помехи средств самой линии. Наименее помехоустойчивыми являются ради-

олинии, хорошей устойчивостью обладают кабельные линии и отличной – волоконно-оптические.

Перекрёстные наводки на ближнем конце линии имеют смысл только в том случае, когда используются несколько линий связи одновременно, в этом случае одна линия может оказывать влияние на другую. Поэтому перекрёстные наводки на ближнем конце определяют степень помехоустойчивости линии к помехам, создаваемым одной линией, на другую. Если ко второй линии будет подключен приёмник, то он может принять наведенную помеху за полезный сигнал.

В связи с тем, что в некоторых новых технологиях используется передача данных одновременно по нескольким линиям, то в последнее время стал применяться показатель, отражающий суммарную мощность перекрёстных наводок от всех передающих линий.

Пропускная способность линии характеризует максимально возможную скорость передачи данных по линии связи. Пропускная способность измеряется в битах в секунду – бит/с, а также в производных единицах, таких как килобит в секунду (Кбит/с), мегабит в секунду (Мбит/с), гигабит в секунду (Гбит/с) и т. д.

Пропускная способность линии связи зависит не только от её характеристик, но и от спектра передаваемых сигналов. Если значимые гармоники сигнала (то есть те гармоники, амплитуды которых вносят основной вклад в результирующий сигнал) попадают в полосу пропускания линии, то такой сигнал будет хорошо передаваться данной линией связи (рис. 3.3*а*).

Если же значимые гармоники выходят за границы полосы пропускания линии связи, то сигнал будет значительно искажаться (рис. 3.3*б*).

Достоверность передачи данных характеризует вероятность искажения линией передаваемых бит информации.

Достоверность передачи данных для разных линий связи может значительно отличаться. Так, для линий связи без дополнительных средств защиты от ошибок она составляет, как правило, величину $10^{-4} – 10^{-6}$, а в оптоволоконных линиях связи доходит до – 10^{-9}. Значение достоверности передачи данных говорит о том, что на указанное количество переданных бит один бит будет искажён.

Искажения происходят как из-за наличия помех на линии, так и по причине искажений формы сигнала, связанной с наличием сопротивлений разной природы и ограниченной полосой пропускания линии.

Удельная стоимость характеризует стоимость единицы длины линии в денежном эквиваленте. Можно создать линию передачи данных с хорошей

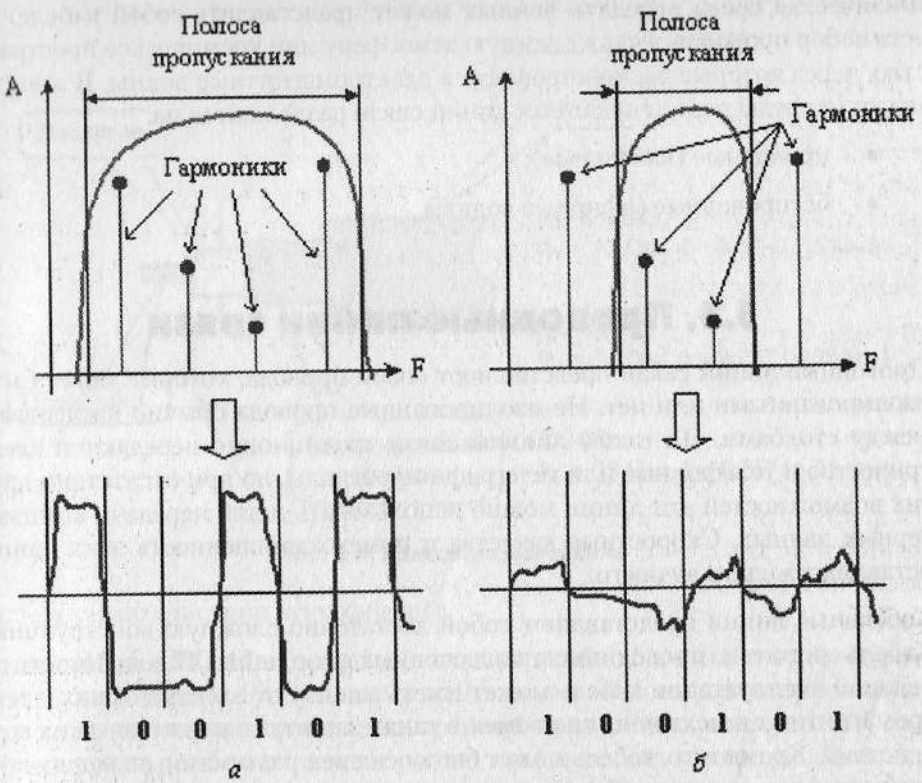


Рис. 3.3. ***Соответствие между полосой пропускания линии и спектром сигнала***

амплитудно-частотной характеристикой, обладающей широкой полосой пропускания, с очень малым затуханием, с высокой помехоустойчивостью, с крайне низкими перекрёстными наводками, с очень высокой пропускной способностью и с высокой достоверностью передачи данной, но немыслимой цены. Поэтому параметр удельной стоимости в большинстве случаев имеет первостепенное значение. В связи с этим разработчики в рамках отведённых средств стремятся обеспечить наиболее значимые показатели для данной линии, это обстоятельство способствует тому, что количество существующих линий связи достаточно велико.

Например, для связи радиолокаторов, стрельбовых комплексов и центра обработки информации, первой в мире системы противоракетной обороны, для обеспечения заданных техническим заданием параметров, в 1958 году была выбрана радиорелейная линия, которая обеспечивала заданные параметры при минимальной стоимости из всех существовавших на тот момент линий связи.

Физическая среда передачи данных может представлять собой кабель, то есть набор проводов, а также земную атмосферу или космическое пространство, через которые распространяются электромагнитные волны. В зависимости от среды передачи данных линии связи разделяются на:

- проводные (кабельные);
- беспроводные (эфирные волны).

3.3. Проводные линии связи

Проводные линии связи представляют собой провода, которые могут быть изолированными или нет. Не изолированные провода обычно натягивают между столбами. По таким линиям связи традиционно передаются электричество и телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии можно использовать и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать лучшего.

Кабельные линии представляют собой достаточно сложную конструкцию. Кабель состоит из проводников, заключенных в изоляцию. В зависимости от условий эксплуатации кабель может иметь защиту от электрических, электромагнитных и механических помех, а также защиту от климатических воздействий. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного пода оборудования. В компьютерных сетях применяются три основных типа кабеля: коаксиальные кабели, кабели на основе скрученных пар проводов, а также волоконно-оптические кабели.

Коаксиальный кабель

Коаксиальный кабель (в переводе означает – имеющий общую ось) состоит из медной жилы, изолятора – окружающего жилу, экрана в виде металлической оплётки и внешней оболочки (рис. 3.4). Экранирующая оплётка защищает передаваемые по кабелю данные, поглощая электромагнитные волны, она должны быть обязательно заземлена с одной стороны. Внешняя оболочка предохраняет кабель от механических повреждений.

Существует большое количество типов коаксиальных кабелей, используемых в сетях различного типа – телефонных, телевизионных и компьютерных. Основным достоинством коаксиального кабеля является малое затухание, что позволяет на его основе строить длинные линии – до нескольких километров со скоростью передачи в сотни МГц. Ниже приведены основные

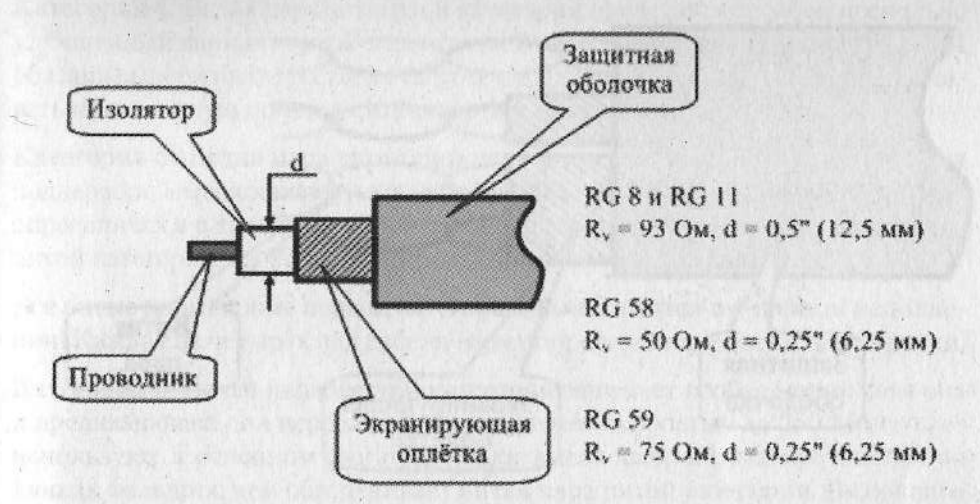


Рис. 3.4. Коаксиальный кабель

типы и характеристики этих кабелей:

- **RG-8 и RG-11** – "толстый" коаксиальный кабель имеет волновое сопротивление 93 Ом и диаметр 0,5 дюйма (12,5 мм). Этот кабель имеет достаточно толстый внутренний проводник диаметром 2,17 мм, который обеспечивает хорошие механические и электрические характеристики (затухание на частоте 10 МГц – не хуже 18 дБ/км). Зато этот кабель сложно монтировать – он плохо гнется;
- **RG-58** – "тонкий" коаксиальный кабель имеет волновое сопротивление 50 Ом. Тонкий внутренний проводник 0,89 мм не так прочен, зато имеет гораздо большую гибкость, удобен при монтаже. Затухание в этом типе кабеля выше, чем в "толстом" коаксиальном кабеле, что приводит к необходимости уменьшать длину кабеля для получения одинакового затухания. Тонкий коаксиальный кабель выпускается в нескольких модификациях: со сплошным или многожильным сердечником.
- **RG-59** – телевизионный кабель с волновым сопротивлением 75 Ом широко применяется в телевидении.

Для передачи данных в условиях сильных электромагнитных или электростатических помех выпускаются кабели с несколькими экранирующими слоями. Основным недостатком коаксиального кабеля является высокая погонная ёмкость, что приводит к сильным искажениям цифровых сигналов.

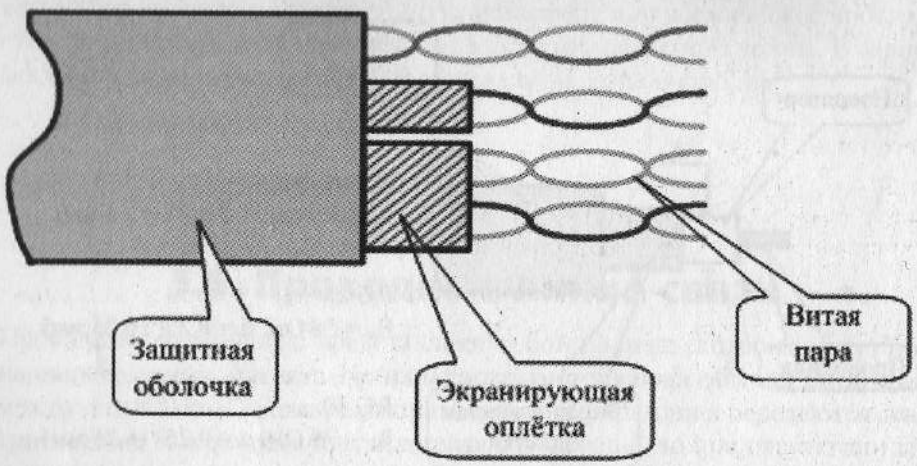


Рис. 3.5. ***Витая пара***

Витая пара

Витая пара представляет собой два перевитых вокруг друг друга изолированных медных провода. Несколько витых пар помещают в одну защитную оболочку. Завивка проводов позволяет избавиться от электрических помех, наводимых соседними парами и другими источниками. Для повышения помехозащищённости витые пары помещают в защитный экран по одной, группами или все. Количество пар в одном экране может быть разным, как это показано на рис. 3.5.

Витые пары в зависимости от электрических и механических характеристик подразделяется на 5 категорий.

Категория 1. Витая пара первой категории не имеет завивки и представляет собой два идущих параллельно проводника. Ярким примером витой пары первой категории является телефонный провод. Скорость передачи информации по такой паре не превышает 64 Кбит/с.

Категория 2. Витая пара второй категории состоит из четырёх витых пар. Это была одна из первых витых пар. Она способна была передавать информацию со скоростью до 4 Мбит/с.

Категория 3. Витая пара третьей категории состоит из четырёх пар с примерно 3 витками на 1 фут (30,5 см), была стандартизована в 1991 году. Стандартом определялись электрические характеристики и скорость передачи до 16 Мбит/с.

Категория 4. Витая пара четвёртой категории представляет собой несколько улучшенный вариант витой пары третьей категории. Витая пара категории 4 обязана выдерживать тесты на частоте передачи сигнала в 20 МГц, обеспечивать повышенную помехоустойчивость и низкие потери сигнала.

Категория 5. Витая пара категории пять была специально разработана для поддержки высокоскоростных протоколов. Поэтому её характеристики определяются в диапазоне до 100 МГц. На заводах изготовителях витая пара пятой категории проходит тестирование на частоте 125 Мбит/с.

Все витые пары, кроме первой категории, выпускаются в 4-парном исполнении. Каждая из четырёх пар кабеля имеет определённый цвет и шаг скрутки.

Категория 6. Витая пара шестой категории занимает особое место, хотя она и предназначена для передачи информации со скоростью до 250 Мбит/с, её используют в основном для поддержки высокоскоростных протоколов на длинах больших, чем обеспечивает витая пара пятой категории. Витая пара шестой категории выпускается в нескольких вариантах: не экранированная, экранированная, частично экранированная.

Категория 7. Витая пара седьмой категории представляет собой улучшенный вариант витой пары шестой категории. Улучшения заключаются в использовании более толстой медной жилы и двойного экранирования, при этом каждая пара имеет собственный экран, а затем все пары помещаются ещё в один общий экран. Это позволяет передавать информацию со скоростью до 600 Мбит/с, однако на практике витая пара седьмой категории используется вместо витой пары пятой категории для передачи информации на большие расстояния или как среда передачи данных, имеющая более высокую достоверность передачи информации. Наличие заземляемого экрана удорожает кабель и усложняет его прокладку, так как требует выполнения качественного заземления.

Волоконно-оптический кабель

Волоконно-оптический кабель представляет собой светопровод (оптическое волокно), выполненный из стекла или прозрачного полимерного материала (в определённом диапазоне длин волн), окруженного другим слоем стекла (полимера), обладающего меньшим показателем преломления, чем светопровод, помещённый в защитную оболочку. Распространяясь по сердечнику, лучи света не выходят за его пределы, отражаясь от покрывающего слоя. Поскольку светопровод имеет очень малый размер – 5-150 мкм, то получение стеклянных волокон такого размера весьма затруднительно, поэтому в настоящее время, благодаря успехам в области химии, в качестве светопровода используются различные полимеры. Защитная оболочка играет очень

важную роль в волоконно-оптическом кабеле. Так как сам светопровод обладает крайне низкими прочностными механическими характеристиками, то защитить светопровод от механических воздействий и призвана защитная оболочка.

В последнее время волоконно-оптический кабель стали называть более скромно – оптическим, но появилось и жаргонное выражение – «оптика». В зависимости от способа распространения света внутри светопровода волоконно-оптические кабели разделяют на два вида: одномодовый кабель и многомодовый кабель.

Одномодовый оптический кабель. Понятие «мода» для оптического кабеля следует понимать как волну света определённой длины. Так, если в кабеле распространяется одна волна, то он будет называться одномодовым, однако для одномодового кабеля не это главное. Этот тип кабеля имеет диаметр светопровода 5-10 мкм, что сравнимо с длиной волны света, поэтому свет в светопроводе распространяется вдоль оптической оси кабеля без внутренних отражений от боковой поверхности (рис 3.6). В качестве светопровода используется полимерный материал, который имеет очень малое затухание в узком диапазоне длин волн. Поэтому для обеспечения большой дальности передачи целесообразно всю энергию источника света сосредоточить только на одной волне.

В качестве источников света используются полупроводниковые квантовые генераторы когерентного излучения – лазеры. Лазерные излучатели работают на длинах волн 1,3 и 1,55 мкм. Быстродействие современных лазеров позволяет модулировать световой поток с частотами до 40 ГГц. Одномодовый

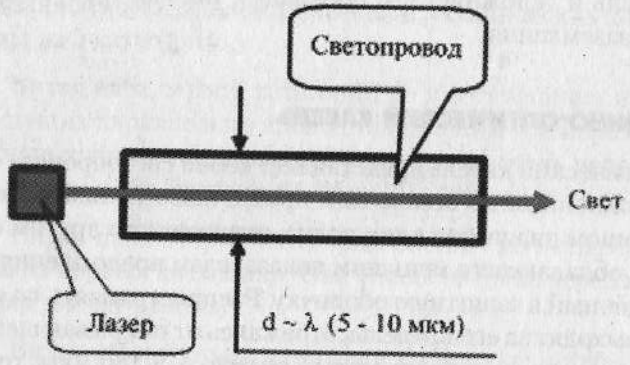


Рис. 3.6. Одномодовый оптический кабель

кабель позволяет передать информацию со скоростью в сотни ГГц на расстояния несколько десятков километров.

Многомодовый оптический кабель. В многомодовых оптических кабелях используется светопровод большего диаметра, который легче изготовить технологически. В светопроводе многомодового кабеля одновременно существует несколько световых лучей, отражающихся от отражателя под разными углами (рис. 3.7).

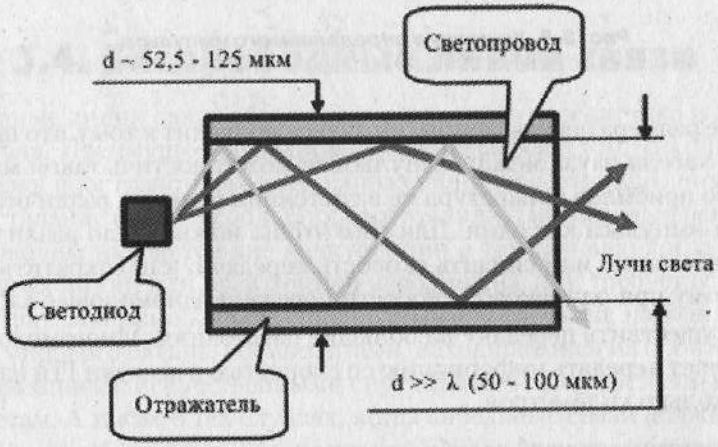


Рис. 3.7. Многомодовый оптический кабель

В качестве источника света для многомодовых оптических кабелях используется полупроводниковый светодиод, излучающий несколько близко расположенных по длине волн. В стандартах определены два наиболее употребительных многомодовых кабеля: 62,5/125 мкм и 50/125 мкм, где 62,5 мкм или 50 мкм – это диаметр светопровода, а 125 мкм – диаметр внешнего отражателя. Светодиоды могут излучать свет с длиной волны 0,85 и 1,3 мкм. Излучатели с длиной волны 0,85 мкм дешевле излучателей с длиной волны 1,3 мкм, но полоса пропускания у них уже – 200 МГц/км вместо 500 МГц/км.

Поскольку в светопроводе одновременно распространяется несколько световых лучей, и они имеют различные углы отражения, в соответствии с длиной волны, то каждый луч внутри светопровода проходит разное расстояние от входа в светопровод до выхода из него. В результате на выходе кабеля, при условии одинаковой скорости распространения света в светопроводе, лучи будут появляться в разное время, какой-то раньше, а какой-то позже. Это приводит к увеличению ширины передаваемого импульса, как показано на рис. 3.8.

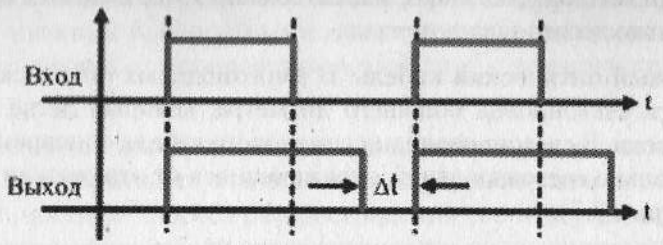


Рис. 3.8. Уширение передаваемого импульса

Увеличение размера передаваемого импульса приводит к тому, что при большой длине кабеля пауза между импульсами может достичь такой малой величины, что приёмная аппаратура не в состоянии будет их различить и воспримет два импульса как один. Для того чтобы можно было различать импульсы, необходимо или снизить скорость передачи, или сократить расстояние. Поэтому при одинаковой скорости передачи одномодовые кабели позволяют осуществить передачу на большие расстояния. Многомодовый кабель позволяет передать информацию со скоростью в десятки ГГц на расстояния в несколько километров.

Волоконно-оптические кабели обладают отличными характеристиками всех типов: электромагнитными, механическими (хорошо гнутся, а в соответствующей изоляции обладают хорошей механической прочностью). Однако у них есть один серьезный недостаток – сложность соединения волокон с разъемами и между собой при необходимости наращивания длины кабеля.

Проблема соединения кабеля с разъёмами решается очень просто в заводских условиях. А вот наращивание кабеля требует крайне дорогостоящего оборудования. Соединение оптических волокон требует проведения высокоточной обрезки волокон в плоскости строго перпендикулярной оси волокна, а также выполнения соединения путём сложной операции сварки, поэтому проведение монтажных работ с оптическим волокном обходится намного дороже из-за трудоемкости операций и высокой стоимости применяемого монтажного оборудования. Выполнение некачественных соединений резко сужает полосу пропускания волоконно-оптических кабелей.

Стоимость волоконно-оптических кабелей практически сравнялась со стоимостью кабелей на витой паре, однако до недавнего времени оконечное оборудование – преобразователи света в электрический сигнал и наоборот – стоило очень дорого. В настоящее время этот разрыв в стоимости оконечного оборудования резко сократился и продолжает сокращаться, поэтому

оптические линии связи перешли из разряда экзотики в разряд рядовых не очень дорогих линий связи.

Несомненным преимуществом волоконно-оптического кабеля является его высокая скорость передачи данных до 40 Гбит/с на большие расстояния, и к тому же он лучше других обеспечивает защиту данных от внешних электромагнитных и электростатических помех (светопровод представляет собой диэлектрик).

3.4. Беспроводные линии связи

Беспроводные линии связи образуются с помощью передатчика и приёмника радиоволн. Преимущество радиоволн заключается в том, что они распространяются в пространстве во всех направлениях и с помощью них возможно осуществление связи на больших расстояниях с минимальными затратами. Радиоволны находятся вне конкуренции в тех случаях, когда прокладка проводных линий или практически невозможна, или требует астрономических затрат. Например, связь с подводными кораблями, находящимися в глубинах мирового океана, космонавтами, находящимися на орбите, или обмен информацией с искусственными спутниками земли, посланными к другим планетам. А также в тех случаях, когда кабельные связи должны проходить через малонаселенную местность или же для связи с мобильным пользователем сети, таким как шофер грузовика, врач, совершающий обход, и так далее.

Сигналы, поступающие от источника (микрофона, телекамеры), как правило, не могут быть непосредственно переданы по радиоканалу. Дело заключается не только в том, что эти сигналы недостаточно велики по амплитуде. Гораздо более существенное обстоятельство заключено в их относительной низкочастотности. Для того чтобы осуществить эффективную передачу сигналов в какой-либо среде с помощью радиоволн, необходимо перенести спектр этих сигналов из низкочастотной области в диапазон достаточно высоких частот.

Идея способа, позволяющего переносить спектр сигнала в область высоких частот, заключается в том, что в передатчике формируется вспомогательный высокочастотный сигнал, называемый несущей частотой. В радиотехнике получили широкое распространение системы, использующие в качестве несущего простое гармоническое колебание. Гармоническое колебание описывается уравнением вида:

$$v_0(t) = V \cos(\omega t + \varphi).$$

В гармоническом колебании возможно изменение трёх свободных параметров: амплитуды – V, частоты – ω и фазы – φ по закону передаваемого сигнала.

Метод изменения одного из параметров несущей частоты полезной информацией во времени называется модуляцией. В радиотехнике используется три основных метода модуляции: амплитудная, частотная и фазовая.

Амплитудная модуляция. Метод наложения полезного сигнала на несущую частоту при амплитудной модуляции определён самим названием. Под воздействием полезного сигнала изменяется амплитуда несущей частоты. На рисунке 3.9 показаны различные методы модуляции. В качестве модулирующего сигнала используется цифровой сигнал – последовательность нолей и единиц. Метод амплитудной модуляции очень просто реализовать, а уж декодирование и подавно. Но он имеет один существенный недостаток. Дальность передачи определяется амплитудой меньшего сигнала. В противном случае приёмник может потерять несущую частоту. Получается, что при достаточно мощном передатчике дальность передачи заведомо меньше, чем могла бы быть. Для хорошей распознаваемости модулирующего сигнала несущая частота должна быть не менее чем в 500 раз выше.

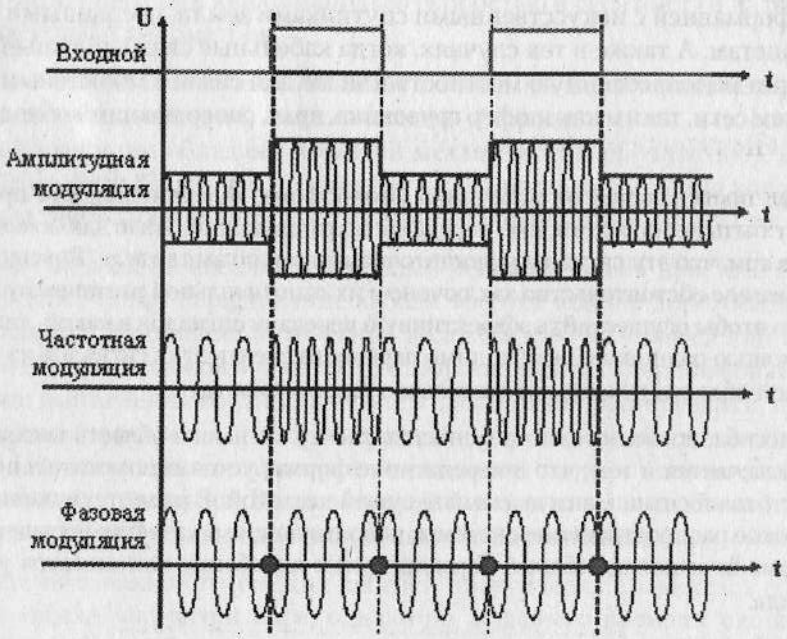


Рис. 3.9. Различные типы методов модуляции

Частотная модуляция. Для увеличения дальности передачи был разработан метод частотной модуляции. В этом случае амплитуда несущей частоты постоянна, а изменяется её частота в небольших пределах, это обусловлено тем, что все колебательные контуры приёмо-передающей аппаратуры имеют хоть и очень узкую, но все же не идеальную амплитудно-частотную характеристику. Поэтому могут реагировать на некоторый диапазон близко расположенных частот. При использовании метода частотной модуляции для качественного распознавания модулирующего сигнала характерно соотношение частот полезного и несущего сигналов в пределах 1:5000.

Фазовая модуляция. Принцип фазовой модуляции основан на возможности изменения фазы несущей частоты. При фазовой модуляции значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но с различной фазой, например 0 и 180 градусов или 0, 90 или 180 и 270 градусов. Это самый перспективный метод модуляции, поскольку соотношение частот полезного сигнала к несущему в идеале может быть 1:1, но и самый сложный в реализации. С помощью фазовой модуляции в волоконно-оптических кабелях теоретически можно достичь скорости передачи в сотни Тбит/с.

Существуют и комбинированные методы модуляции, например амплитудно-частотная или частотно-фазовая, но они применяются только в радиолокации, что позволяет более качественно производить селекцию малоразмерных целей.

Импульсная модуляция

В радиотехнике очень часто в качестве переносчика полезной информации используют периодическую последовательность сравнительно узких импульсов. Последовательность импульсов одного знака $v_0(t)$ характеризуется параметрами (рис. 3.10): амплитудой импульсов V; длительностью (шириной) импульсов τ; частотой следования (или тактовой частотой) импульсов

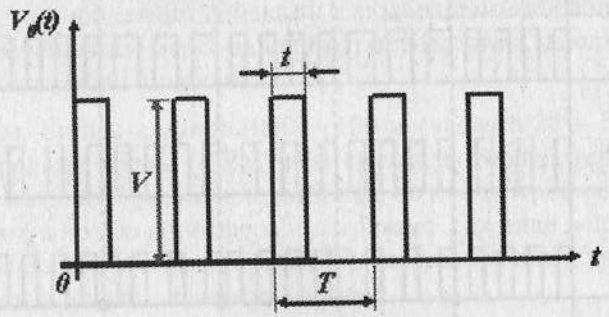


Рис. 3.10. Периодическая последовательность узких импульсов

$f_T = 1/T$, где T – период следования импульсов ($\omega_T = 2\pi f_T$); положением (фазой) импульсов относительно тактовых (отчётных) точек. Отношение T/τ называется скважностью импульсов.

По закону передаваемого полезного сигнала можно изменять (модулировать) любой из перечисленных параметров импульсной последовательности. При этом модуляция называется импульсной.

В зависимости от того, какой параметр модулируется полезным сигналом, различают:

- *амплитудно-импульсную* модуляцию (АИМ), когда по закону полезного сигнала изменяется амплитуда импульсов;
- *широтно-импульсную* модуляцию (ШИМ), когда изменяется ширина импульсов;
- *частотно-импульсную* модуляцию (ЧИМ), когда изменяется частота следования импульсов;
- *фазо-импульсную* модуляцию (ФИМ), когда изменяется фаза импульсов, то есть временное положение относительно тактовых точек.

Перечисленные виды модуляции представлены на рис. 3.11. Модуляцию ФИМ и ЧИМ объединяют во временно-импульсную (ВИМ) модуляцию. Между ними существует связь, аналогичная связи между фазовой и частотной модуляцией синусоидального сигнала.

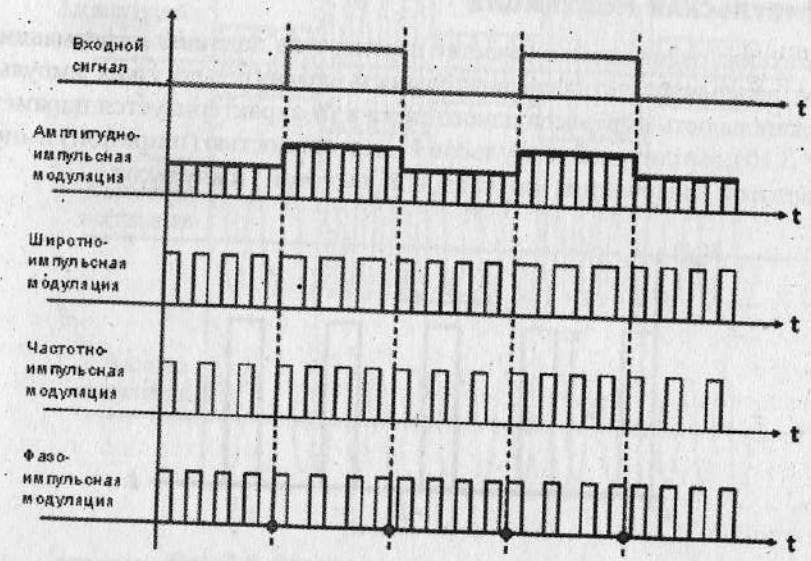


Рис. 3.11. Виды импульсной модуляции

Импульсные последовательности, представленные на рис. 3.11, называются последовательностями видеоимпульсов. Если позволяет среда распространения, то видеоимпульсы передаются без дополнительных преобразований, например, по кабелю. Однако по радиолиниям передать видеоимпульсы невозможно, поэтому их подвергают преобразованию – модуляции.

Модулируя с помощью видеоимпульсов гармоническое несущее колебание достаточно высокой частоты, получают радиоимпульсы, которые способны распространяться в окружающем пространстве. Полученные в результате сочетания первой и второй ступени модуляции сигналы могут иметь названия АИМ–АМ, ФИМ–АМ, ФИМ–ЧМ и другие.

Сравнение импульсных видов модуляции показывает, что АИМ имеет меньшую ширину спектра по сравнению с ШИМ и ФИМ. Однако последние более устойчивы к воздействию помех. Для обоснования выбора метода модуляции в системах передачи необходимо сравнить эти методы по различным критериям: энергетическим затратам на передачу сигнала, помехоустойчивости, сложности оборудования и др.

Диапазоны частот

В зависимости от длины волны несущей частоты радиоволны обладают разными характеристиками, поэтому радиоволны подразделяются на диапазоны в зависимости от частоты несущего сигнала. Существует 7 основных диапазонов частот.

Сверхдлинные и длинные волны. Радиоволны длиной 100 – 10 км (частота 3 – 30 КГц) и длиной 10 – 1 км (частота 30 – 300 КГц) называются сверхдлинными (СДВ) и длинными (ДВ) волнами соответственно. Они распространяются в свободном пространстве вдоль поверхности Земли днём и ночью. Отличительной особенностью этих длин волн является их возможность распространяться в воде. Поэтому на таких волнах осуществляется односторонняя связь центра управления войсками с подводными кораблями. Однако они сильно ослабевают по мере удаления от передатчика, и поэтому передатчики должны быть очень мощными.

Средние волны. Волны длиной 1000 – 100 м (частота 30 – 300 КГц), так называемые средние волны (СВ), днём сильно поглощаются ионосферой (верхним слоем атмосферы, имеющим большую концентрацию ионов) и быстро ослабевают, а ночью ионосфера их отражает. Средние волны используют для радиовещания, причём днём можно слышать только близкорасположенные станции, а ночью – очень удалённые.

Короткие волны. Волны длиной 100 – 10 м (частота 3 – 30 МГц), называемые короткими волнами (КВ), приходят к приёмнику, отражаясь от ионосферы, причём днём отражаются более короткие, а ночью – более длинные из них. Для таких радиоволн можно создавать антенны передатчиков, которые излучают электромагнитную энергию направленно, фокусируя её в узкий луч, и таким образом увеличивать мощность сигнала, идущего к антенне приёмника. На коротких волнах работает большинство станций радиосвязи – судовых, самолётных и т.д., а также многие радиовещательные станции. Диапазон коротких волн подразделяется на более узкие диапазоны, называемые поддиапазонами. При всех достоинствах коротких волн, на них влияет движение верхних слоёв атмосферы, поэтому после отражения от ионосферы происходит изменение несущей частоты сигнала, как в сторону увеличения, так и уменьшения, в соответствии с направлением движения ионосферы. Поэтому приёмники должны обладать возможностью подстраиваться под принимаемую частоту, это приводит с одной стороны к усложнению приёмников, а с другой к ухудшению разборчивости речи и помехам.

Ультракороткие волны. Радиоволны длиной 10 – 0,3 м (частота 30 – 1000 МГц), называются ультракороткими волнами (УКВ). Эти волны не поглощаются и не отражаются ионосферой, а пронизывают её и уходят в космос. Поэтому связь на УКВ волнах возможна только на таких расстояниях, когда антенна приёмника «видит» антенну передатчика, то есть в пределах прямой видимости, когда ничего нет между антеннами – гора, выпуклость Земли, дом. Поэтому УКВ диапазон используется в основном для радиорелейной связи и телевидения. Земля хоть и очень большая, но всё-таки круглая, поэтому её кривизна сказывается на пределах прямой видимости, например, при высоте антенны в 1 м линия горизонта будет находиться на расстоянии 5 км на уровне моря. Для увеличения зоны уверенного приёма антенну передатчика поднимают как можно выше.

Сверхвысокочастотные волны. Радиоволны длиной менее 3 см (частота более 10 ГГц) называются сверхвысокочастотными. Эти волны не поглощаются атмосферой, поэтому на них осуществляется связь с космическими аппаратами, зато они хорошо поглощаются тканями живых организмов. На этих волнах работают все СВЧ-печки. Поэтому находиться в пределах работы станции космической связи не рекомендуется.

Инфракрасные волны. При достижении радиоволнами частоты в 100 ТГц и более они переходят в световой диапазон. Самый нижний диапазон световых волн – инфракрасные волны, они не видимы глазом человека, но поглощаются и отражаются всеми предметами, находящимися на их пути. Поэтому на этих волнах возможна устойчивая передача на небольшие расстояния в пределах прямой видимости. Инфракрасные волны нашли широкое применение в пультах дистанционного управления бытовыми электро- и радиоприборами.

Существует большое количество различных типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. Диапазоны коротких, средних и длинных волн (КВ, СВ и ДВ), называемые также диапазонами амплитудной модуляции (АМ) по типу используемого в них метода модуляции сигнала, обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн (УКВ), для которых характерна частотная модуляция (Frequency Modulation, FM), а также на диапазонах сверхвысоких частот (СВЧ). В диапазоне СВЧ (свыше 4 ГГц) сигналы уже не отражаются ионосферой Земли и для устойчивой связи требуется наличие прямой видимости между передатчиком и приёмником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

В компьютерных сетях передачи данных в настоящее время применяются практически все описанные типы физических сред передачи данных, но наиболее перспективными являются волоконно-оптические: на них строятся как магистрали крупных территориальных сетей, так и высокоскоростные линии связи локальных сетей. Популярной средой является также витая пара, которая характеризуется отличным соотношением качества к стоимости, а также простотой монтажа. Спутниковые каналы и радиосвязь используются чаще всего в тех случаях, когда кабельные связи применить невозможно.

3.5. Объединение компьютеров в сеть

При объединении в сеть большего числа компьютеров возникает целый комплекс проблем. Любая сеть передачи данных описывается двумя понятиями: топологией и архитектурой. Топология определяет взаимное расположение компонент сети относительно друг друга в пространстве. Архитектура же определяет способы и методы передачи информации по сети.

Прежде всего, необходимо рассмотреть топологию сети, поскольку конфигурация физических связей, определяемая электрическими соединениями компьютеров между собой, существенно влияет на многие характеристики сети. Рассмотрим наиболее часто встречающиеся топологии.

Общая шина (рис. 3.12) является очень распространённой (а до недавнего времени самой распространённой) топологией для локальных сетей. В этом случае компьютеры (узлы) подключаются к одному каналу связи. Передаваемая информация может распространяться в обе стороны. Применение общей

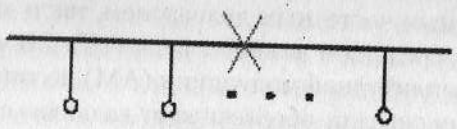


Рис. 3.12. Топология общая шина

шины снижает стоимость линии связи, унифицирует подключение различ-
ных модулей, обеспечивает возможность почти мгновенного широковеща-
тельного обращения ко всем узлам сети. Таким образом, основными преиму-
ществами такой топологии являются дешевизна и простота. Самый серьёз-
ный недостаток общей шины заключается в её низкой надежности: любой
дефект линии связи полностью парализует всю сеть. Другим недостатком
этой топологии является её невысокая производительность, так как при та-
ком способе подключения в каждый момент времени только один компью-
тер может передавать данные в сеть. Поэтому пропускная способность кана-
ла связи всегда делится между всеми узлами сети.

Топология звезда (рис. 3.13). Для организации звезды необходимо некото-
рое центральное устройство, к которому индивидуальными каналами связи
подключаются узлы. Такое устройство принято называть концентратором
или повторителем, в простейшем случае концентратор может представлять
собой набор клемм, к которым механически подключаются электрические
каналы связи (кабели). Главное преимущество этой топологии – существен-
но большая надежность. Любые проблемы с каналом связи касаются лишь
одного узла, а не всей сети, а вот неисправность концентратора может выве-
сти из строя всю сеть. К недостаткам этой топологии следует отнести нали-
чие большого количества каналов связи и необходимость приобретения кон-
центратора. Кроме того, возможности по наращиванию количества узлов в
сети ограничиваются количеством портов концентратора.

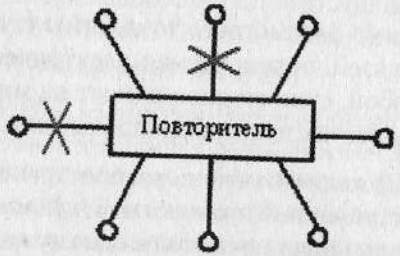


Рис. 3.13. Топология звезда

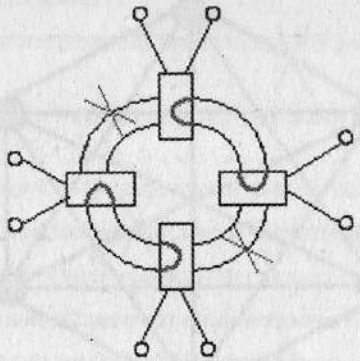


Рис. 3.14. Топология кольцо

Топология кольцо. В сетях с кольцевой конфигурацией (рис. 3.14), как и в топологии звезда, используются специальные устройства (концентраторы) для организации кольца, а уже к ним индивидуальными каналами связи подключаются узлы.

Данные передаются по кольцу от одного концентратора к другому, как правило, в одном направлении. Если узел определяет, что данные адресованы ему, то он копирует их себе во внутренний буфер. В сети с кольцевой топологией выход из строя или отключение какого-либо узла не влияет на работу сети. Для повышения надёжности кольцевой топологии применяют два кольца. В случае выхода из строя одного кольца задействуется второе кольцо, в результате чего топология сохраняет работоспособность, однако длина кольца увеличивается в два раза. При выходе из строя двух каналов связи между концентраторами кольцо распадается на полукольца.

Кольцо представляет собой очень удобную конфигурацию для организации обратной связи: данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому этот узел может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно.

Полносвязанная топология. В полносвязанной топологии (рис. 3.15) каждый узел индивидуальными каналами связи связан со всеми узлами сети. Достоинством полносвязанной топологии является высочайшая надёжность, поскольку выход из строя одного или даже нескольких каналов связи не приводит к выходу сети из строя. Обратной стороной высокой надёжности является большое количество каналов связи, а следовательно, высокая стоимость. Поэтому такие топологии целесообразно применять в тех

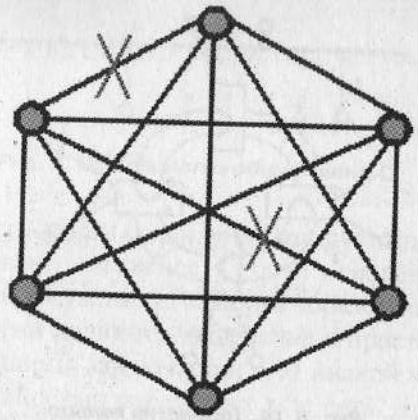


Рис. 3.15. Полносвязанная топология

случаях, когда требуется обеспечить высочайшую надёжность сети, напри-мер в системах управления высокоопасных производств (химических, атом-ных реакторов, атомного топлива), в военных системах.

На практике вышеописанные топологии в чистом виде применяются в весь-ма ограниченных количествах. Реальные сети представляют собой состав-ные сети, состоящие из большого количества различных топологий, соеди-нённых между собой самым что ни на есть причудливым образом. Такие со-единения характерны для глобальных сетей, сетей студенческих городков и городских сетей. Поэтому во многих случаях даже администраторы плохо себе представляют топологию сети, с которой они работают, что приводит к большому количеству проблем по администрированию сети.

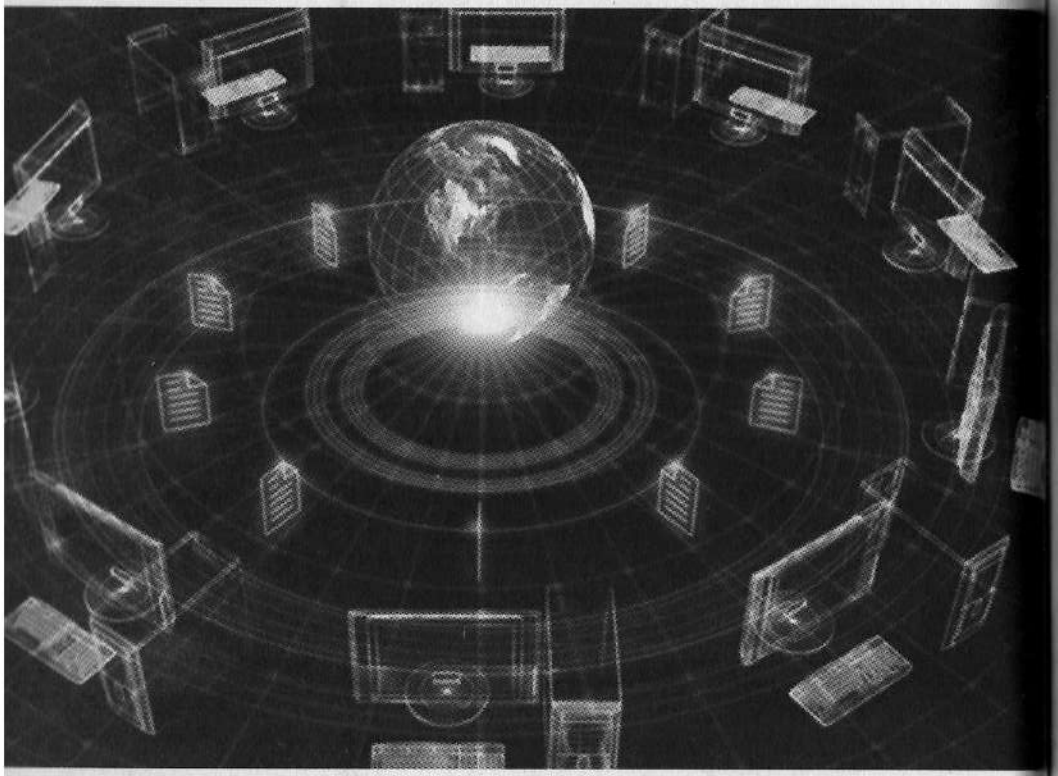
Контрольные вопросы

1. *Какие типы линий связи Вам известны?*
2. *Назовите основные характеристики линий связи.*
3. *Назовите достоинства и недостатки коаксиального кабеля.*
4. *Зачем в коаксиальном кабеле применяется экранирующая оплётка?*
5. *Назовите достоинства и недостатки витой пары.*
6. *Почему витая пара позволяет передавать информацию на меньшие расстояния по сравнению с коаксиальным кабелем?*
7. *Назовите достоинства и недостатки оптического кабеля.*

8. *Перечислите основные характеристики кабеля на основе витой пары известных Вам категорий.*
9. *В чём заключается отличие одномодового оптического кабеля от многомодового?*
10. *Почему дальность передачи у одномодового оптического кабеля выше чем у многомодового?*
11. *Назовите достоинства и недостатки амплитудной модуляции.*
12. *Назовите достоинства и недостатки частотной модуляции.*
13. *Назовите достоинства и недостатки фазовой модуляции.*
14. *Зачем радиоволны разделяют на диапазоны?*
15. *Какими достоинствами обладают сверхдлинные волны?*
16. *Почему все экспедиции используют связь в КВ-диапазоне?*
17. *Какова дальность передачи в УКВ-диапазоне?*
18. *Почему все беспроводные сетевые устройства работают в СВЧ-диапазоне?*
19. *Почему инфракрасный диапазон имеет ограниченное применение?*
20. *Дайте определение понятия топология.*
21. *Назовите достоинства и недостатки топологии общая шина.*
22. *Назовите достоинства и недостатки топологии звезда.*
23. *Назовите достоинства и недостатки топологии кольцо.*
24. *В каких случаях целесообразно применять полносвязанную топологию?*

Глава 4.

МЕТОДЫ ПЕРЕДАЧИ ДАННЫХ



Сети передачи данных связывают между собой узлы, чаще всего в роли узлов выступают компьютеры, поэтому информация между ними передаётся в цифровом виде. Но иногда требуется передавать значения некоторого аналогового сигала, например, при построении систем управления каким-либо объектом необходимо передавать реакцию объекта на внешние и внутренние воздействия, которые представляют собой функцию напряжения во времени по цифровым каналам связи. Для преобразования аналогового сигнала в цифровой используется метод дискретизации. Метод дискретизации заключается в измерении амплитуды аналогового сигнала в определённые моменты времени с некоторым, заранее определённым, шагом и преобразовании измеренных амплитуд в цифровую форму.

На рисунке 4.1 показан фрагмент аналогового сигнала. В моменты времени $t_1 - t_8$ с шагом Δt производится изменение амплитуд сигнала $a_1 - a_8$, которые затем представляются в цифровом виде определенной разрядности, которая обеспечивает представление сигнала с заданной точностью. Устройство, выполняющее преобразование аналогового сигнала в цифровой, называется аналого-цифровым преобразователем (АЦП). После этого замеры передаются по каналам связи в виде последовательности единиц и нулей. Преимуществом цифровых методов передачи аналоговой информации является возможность контроля достоверности полученных по линии связи данных.

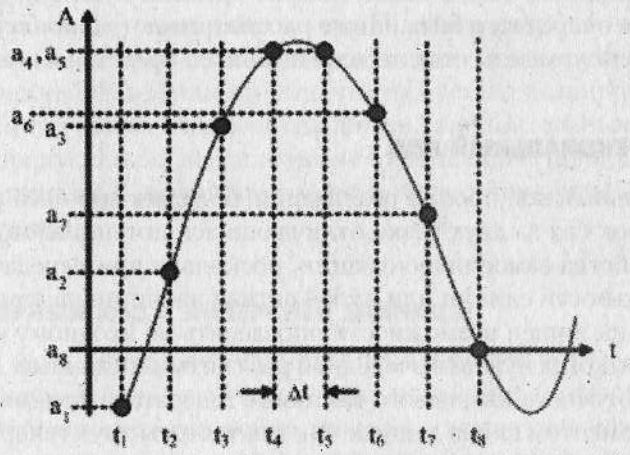


Рис. 4.1. Преобразование аналогового сигнала в цифровую форму

В цифровой технике для представления двоичной информации – нолей и единиц – используют два основных метода (кода): потенциальный и импульсный. В потенциальном коде логические единицы и ноли представляются значением некоторого потенциала, а его перепады, формирующие законченные импульсы, во внимание не принимаются. В импульсном коде логические единицы и ноли представляются либо импульсами определенной полярности, либо частью импульса – перепадом потенциала определенного направления во время действия импульса.

4.1. Коды, применяемые в сетях

Поскольку в сетях передачи данных используется минимальное количество линий связи, то информация передаётся в последовательном виде, без специального канала синхронизации, поэтому необходимо иметь какие-то возможности для того, чтобы приёмник точно знал, в какой момент времени необходимо считывать новую информацию с линии связи. Даже использование канала синхронизации не гарантирует правильного приёма информации вследствие того, что на больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придёт позже или раньше соответствующего сигнала данных, благодаря этому бит данных будет пропущен или считан повторно. Поэтому в сетях применяются самосинхронизирующиеся коды, сигналы которых несут для приёмника указания о том, в какой момент времени нужно осуществлять распознавание очередного бита. Ниже рассматриваются наиболее популярные коды, используемые в сетях передачи данных, представленные на рис. 4.2.

Потенциальный код

Потенциальный код прост в реализации, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но абсолютно лишён свойства самосинхронизации, поскольку при передаче длинной последовательности единиц или нулей сигнал на линии не изменяется, поэтому приёмник лишён возможности определять по входному сигналу моменты времени, когда нужно в очередной раз считывать данные. Даже при наличии высокоточного кварцевого тактового генератора приёмник может ошибиться с моментом съёма данных, так как частоты двух генераторов никогда не бывают полностью идентичными и всегда присутствует сдвиг фаз. Сдвиг фаз выражается в том, что даже если частоты сигналов полностью идентичны

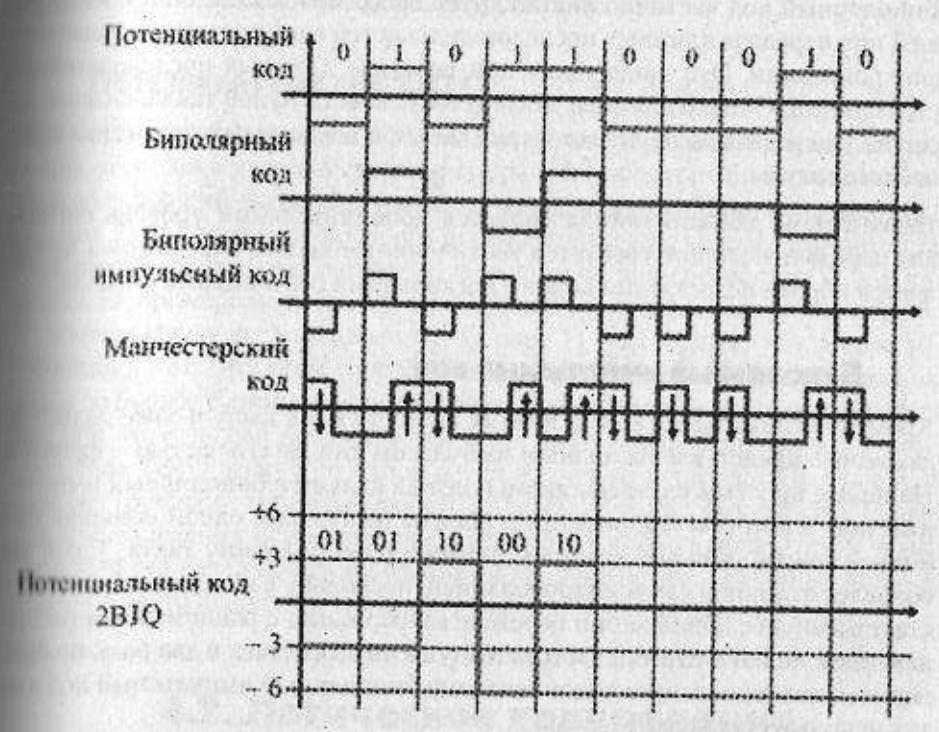


Рис. 4.2. Коды, применяемые в сетях передачи данных

Другим недостатком потенциального кода является наличие низкочастотной составляющей, которая приближается к нулю при передаче длинных последовательностей нулей или единиц, что требует очень широкой полосы пропускания линии связи, а наличие на линии дополнительных устройств, не способных пропускать низкочастотные составляющие, приводит к резкому затуханию сигнала. В результате в чистом виде потенциальный код в сетях не используется.

Биполярный код с инверсией единицы

В биполярном коде с инверсией единицы используются три уровня потенциала – отрицательный, нулевой и положительный. 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

Биполярный код частично ликвидирует проблемы постоянной составляющей при передаче длинных последовательностей единиц и отсутствия самосинхронизации. Это происходит при передаче длинных последовательностей единиц. Однако длинные последовательности нулей также опасны, поскольку передаваемый сигнал вырождается в постоянный потенциал нулевой амплитуды.

Недостатком данного метода является дополнительный уровень сигнала, для передачи которого требуется увеличение мощности передатчика, что является общим недостатком кодов с несколькими состояниями сигнала.

Биполярный импульсный код

Кроме потенциальных кодов, в сетях используются импульсные коды, когда данные представлены полным импульсом или же его частью – фронтом. Наиболее простым случаем такого подхода является биполярный импульсный код, в котором единица представлена импульсом одной полярности, а ноль – другой. Каждый импульс длится ровно половину такта. Такой код обладает отличной самосинхронизацией, поскольку в каждом такте присутствует импульс. Однако при передаче информации с равномерным распределением нолей и единиц частота импульсов возрастает в два раза, по сравнению с потенциальным кодом, поэтому биполярный импульсный код в сетях используется редко.

Манчестерский код

В манчестерском коде для кодирования единиц и нулей используется фронт импульса. При манчестерском кодировании каждый такт делится на две равные части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня к высокому, а ноль – обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Поскольку сигнал изменяется не менее одного раза за такт, то манчестерский код обладает отличной самосинхронизацией. При передаче чередующихся нолей и единиц частота передачи равна частоте передачи потенциального кода, при длинных же последовательностях нолей или единиц частота возрастает в два раза по сравнению с потенциальным кодом. Поэтому манчестерский код сочетает в себе достоинства предыдущих двух рассмотренных кодов. На практике чередующиеся последовательности нолей и единиц встречаются не всегда, как, впрочем, и длинные последовательности нолей и единиц, поэтому средняя частота передачи составляет 3/4 от частоты потенциального кода. До недавнего времени

это был основной код передачи данных в сетях.

Потенциальный код 2B1Q

Для передачи информации по сетям возможно использование кодов, имеющих несколько уровней сигнала. На рис. 4.2 представлен потенциальный код, обладающий четырьмя уровнями сигнала для кодирования данных, – это код 2B1Q, название которого отражает его суть: каждые два бита (2B) передаются за один такт сигналом, имеющим четыре состояния (1Q). В приведённом примере паре бит 00 соответствует потенциал минус 6v, паре бит 01 соответствует потенциал минус 3v, паре 11 – потенциал +6v, а паре 10 – потенциал +3v. При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар бит, так как при этом сигнал превращается в постоянную составляющую. С помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью любого из рассмотренных выше кодов. Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня чётко различались приёмником на фоне помех. Подобные коды достаточно часто используются при построении сетей передачи данных.

4.2. Логическое кодирование

На пропускную способность линии оказывают влияние не только применяемые коды, рассмотренные выше, но и логическое кодирование. Логическое кодирование подразумевает замену бит исходной информации новой последовательностью бит, несущей ту же информацию, но обладающей, кроме этого, дополнительными свойствами, например возможностью для приёмной стороны обнаруживать ошибки в принятых данных. При логическом кодировании чаще всего исходная последовательность бит заменяется более длинной последовательностью, поэтому пропускная способность канала по отношению к полезной информации при этом уменьшается.

Для логического кодирования применяются два основных метода – избыточные коды и скремблирование.

Избыточные коды

Избыточные коды основаны на разбиении исходной последовательности бит на группы, которые часто называют символами. Затем каждый исходный символ заменяется на новый, который имеет большее количество бит, чем исходный.

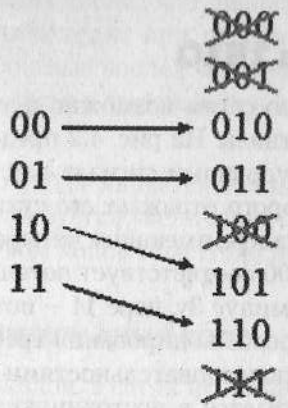


Рис. 4.3. Избыточный код 2В/3В

Например, логический код 2В/3В, приведённый на рисунке 4.3, заменяет исходные символы длиной 2 бита на символы длиной 3 бита. Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Поскольку в коде 2В/3В результирующие символы могут содержать 8 битовых комбина-ций в то время как исходные символы – только 4 то в результирующем коде можно ото-брать 4 таких комбинаций, которые при любых комбинациях полученных символов дают последовательность, содержащую не более трёх нолей и четырёх единиц подряд. Такие комбинации приведены на рис.4.3. Остальные символы можно использовать для передачи служебной информации или считать их запрещёнными кодами. Кроме устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приёмнику распознавать искаженные биты.

Затем код 2В/3В передаётся по линии с помощью физического кодирования по одному из методов потенциального кодирования, чувствительному только к длинным последовательностям нулей. Основная задача разработчиков состоит в выборе кода с меньшей избыточностью, поэтому символы выбираются большого размера. В нашем примере каждые два бита кодируются тремя – избыточность составляет 50%, а скорость передачи падает в 1,5 раза. В современных сетях используются избыточные коды большей длины, например 64В/66В, в этом случае избыточность составляет всего 3,1%.

Буква В в названии кода означает, что элементарный сигнал имеет два состояния (от английского binary – двоичный). Имеются также коды и с тремя состояниями сигнала. Например, в коде 8В/6Т для кодирования 8 бит

исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода 8В/6Т выше, чем кода 2В/3В, так как на 256 исходных кодов приходится $3^6 = 729$ результирующих символов, количество символов меньше, поэтому скорость передачи информации в 1,33 раза выше.

Использование таблицы перекодировки является очень простой операцией, поэтому этот подход не усложняет сетевые адаптеры и интерфейсные блоки передающей и приёмной аппаратуры.

Скремблирование

Одним из методов уменьшения количества идущих подряд нолей или единиц является метод перемешивания, заключающийся в побитном вычислении результирующего кода, на основании бит исходного кода, и полученных в предыдущих тактах бит результирующего кода по определённому закону. В качестве примера разберём работу скремблера, реализующего следующее соотношение:

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$$

где:

- B_i – двоичная цифра результирующего кода, полученная на i-м такте работы скремблера;
- A_i – двоичная цифра исходного кода, поступающая на i-м такте на вход скремблера;
- B_{i-3} и B_{i-5} – двоичные цифры результирующего кода, полученные на предыдущих тактах работы скремблера, соответственно на 3 и на 5 тактов ранее текущего такта;
- $\oplus$ – операция исключающего ИЛИ (сумма по модулю два).

Для исходной последовательности 110110000001 скремблер (рис. 4.4) даст следующий результирующий код 110001101111, в котором нет последовательности из шести нулей, присутствовавшей в исходном коде. Отличительной особенностью скремблеров, основанных на использовании операции «сумма по модулю два», является их обратимость. То есть один и тот же скремблер является кодировщиком и декодировщиком. После получения результирующей последовательности приёмник передает её на вход скремблера, который восстанавливает исходную последовательность на основании обратного соотношения:

$$C_i = B_i \oplus B_{i-3} \oplus B_{i-5}$$

как это показано на рисунке 4.4. Если на вход скремблера подать полученную последовательность (см. сверху), то можно получить на выходе исходную последовательность.

Различные алгоритмы скремблирования отличаются количеством слагаемых, дающих цифру результирующего кода, и сдвигом между слагаемыми.

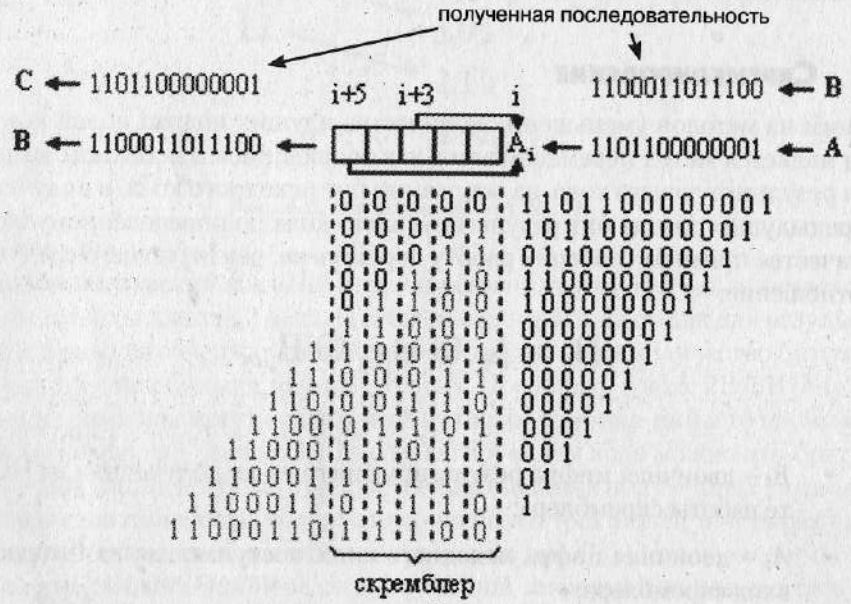


Рис. 4.4. Преобразование последовательности в скремблере

В некоторых случаях для уменьшения количества идущих подряд нолей и единиц возможно использование сразу двух методов: избыточного кодирования и скремблирования. Например, сначала применяется метод избыточного кодирования, а затем скремблирование.

В силу того, что линия связи обладает целым рядом характеристик, ухудшающих передаваемую информацию, и сама информация тоже способствует ухудшению качества приёма, например, длинные последовательности нолей и единиц, то приёмник не всегда может принять её правильно, что приводит к появлению ошибок. Во многих случаях появление ошибок недопустимо, а иногда может привести к катастрофе. Поэтому следует иметь методы обнаружения и исправления (коррекции) ошибок.

4.3. Обнаружение и коррекция ошибок

Вся передаваемая информация перед началом передачи по сети разбивается на группы, которые принято называть пакетами или кадрами. Размер кадра определяется технологией передачи, которая поддерживается соответствующим протоколом. Все методы обнаружения ошибок основаны на передаче в составе кадра данных дополнительной служебной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. Эту служебную информацию принято называть контрольной суммой, хотя контрольная сумма вычисляется как функция от основной информации, причём необязательно только путём суммирования. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному закону и в случае её совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть без ошибок.

Существует несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Контроль чётности или нечётности представляет собой наиболее простой метод контроля данных. Суть метода заключается в том, что к каждому байту добавляется один служебный разряд, значение которого вычисляется по содержимому байта, таким образом, чтобы общее количество единиц в девяти разрядах было чётным – при контроле по чётности или нечётным – при контроле по нечётности, как показано на рисунке 4.5.

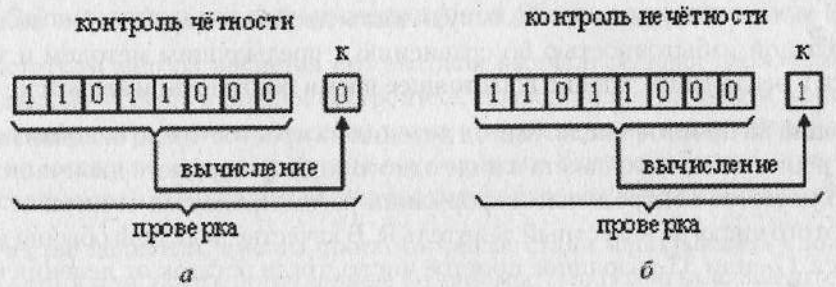


Рис. 4.5. Контроль информации методом: а – четности, б - нечетности

Контроль чётности/нечётности является наименее мощным методом контроля, так как с его помощью можно обнаружить только одиночные ошибки в проверяемых данных, двойные ошибки не приводят к изменению контрольного разряда, поэтому и не определяются. Этот метод обладает крайне большой избыточностью – 12,5% и невысокой диагностической способностью, в силу чего редко применяется в сетях передачи данных.

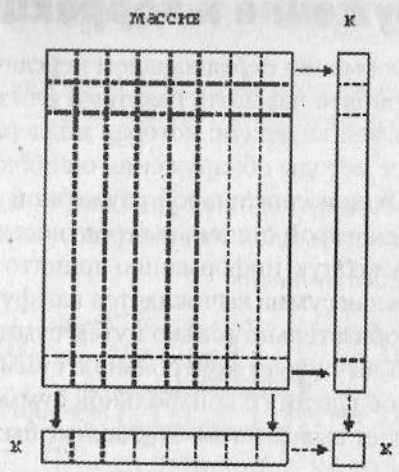


Рис. 4.6. Вертикальный и горизонтальный контроль чётности

Вертикальный и **горизонтальный контроль чётности** представляет собой модификацию описанного выше метода. Его отличие состоит в том, что исходные данные рассматриваются в виде матрицы, строки которой составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца, матрицы. Последний контрольный разряд обычно является контрольным разрядом столбца контрольных разрядов строк (рис. 4.6), хотя может быть и контрольным разрядом строки контрольных разрядов столбцов.

Этот метод обнаруживает большую часть двойных ошибок. Он обладает небольшой избыточностью по сравнению с предыдущим методом и также прост в реализации, однако в настоящее время почти не применяется.

Деление на простое число. Метод деления на простое число основан на рассмотрении исходного пакета в виде одного многоразрядного двоичного числа. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель R. В качестве делителя обычно выбирается 17- или 33-разрядное простое число, тогда остаток от деления будет иметь длину 16 разрядов (2 байта) или 32 разряда (4 байта) соответственно (рис. 4.7). Полученный остаток присоединяется к пакету, в качестве контрольной суммы.

При получении кадра данных снова вычисляется остаток от деления на тот же делитель **R**, но при этом делению подвергается весь пакет, содержащий информацию и присоединённую контрольную сумму. Если остаток от деления на **R** равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

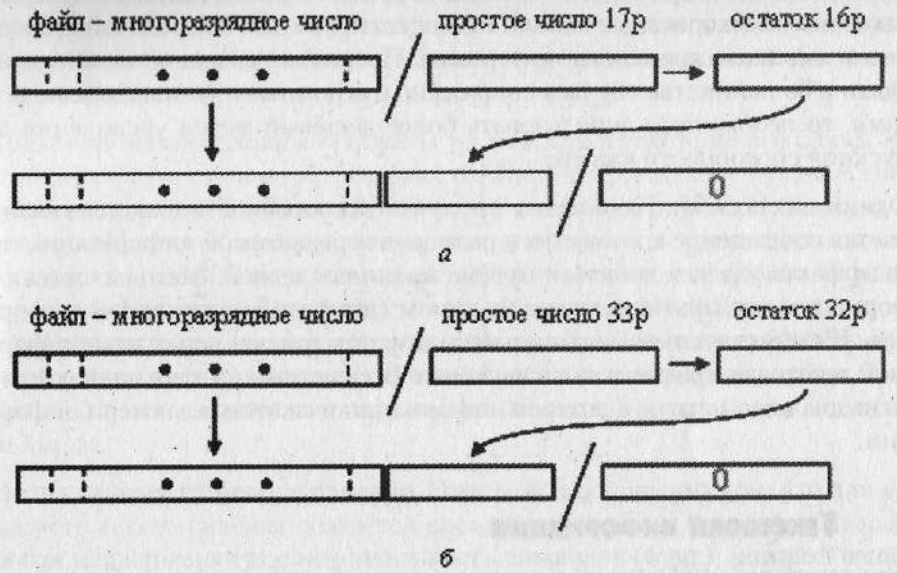


Рис. 4.7. Деление на простое число: а – 17-ти разрядное, б – 32-х разрядное

Метод деления на простое число является в настоящее время наиболее популярным методом контроля в сетях передачи данных, поскольку обладает более высокой диагностической возможностью: он обнаруживает все одиночные, двойные ошибки и ошибки в нечётном числе бит. Метод обладает также невысокой степенью избыточности. Например, для кадра размером 1024 байта контрольная информация длиной 4 байта составляет всего 0,4 %.

Недостатком метода является его высокая вычислительная сложность, но невиданные успехи в микроэлектронике, позволившие на одном кристалле создавать огромное количество транзисторов и на их основе реализовывать колоссально сложные схемы высочайшего быстродействия, приемлемой стоимости, свели на нет все вычислительные недостатки метода.

В качестве делителя, вместо простого числа, стали использовать сложные выражения или даже многочленные полиномы. Это позволило значительно увеличить вероятность обнаружения ошибки.

4.4. Сжатие информации

Современные линии связи имеют очень высокие скорости передачи информации – до десятков Гбит/с, но и количество передаваемой информа-

ции постоянно возрастает, а в последнее время лавинообразно, поэтому даже такие высокоскоростные каналы не могут справиться с передачей информации в заданные временные интервалы. Поскольку создание нового канала связи в большинстве случаев сопряжено с огромными финансовыми затратами, то необходимо использовать более дешёвый метод увеличения пропускной способности канала.

Одним из способов увеличения пропускной способности каналов связи является сокращение количества и размеров передаваемой информации, такого эффекта удаётся добиться путём сжатия последней. Методы сжатия информации неразрывно связаны с типом (природой) сжимаемой информации. В вычислительной технике используется три основных типа информации: текстовая, графическая и звуковая. Все системы сжатия подразделяются на два вида: сжатие с потерей информации и сжатие без потери информации.

Текстовая информация

Для текстовой информации сжатие с потерей информации в большинстве случаев неприемлемо, поскольку такое сжатие приводит к искажению смысла передаваемой информации. Для сжатия текстовой информации используется несколько основных методов, на основе которых разработано большое количество программ-архиваторов.

Метод перекодировки. Под перекодировкой понимается любое удачно выбранное представление исходной информации, при котором её можно легко восстановить обратно и которое занимает меньший физический объём на носителе. Примерами перекодировки являются стенография, нотная запись, использование аббревиатур. Интересным способом сжатия программ является запись программы на языке высокого уровня, поскольку её текст обычно меньше размера исполняемого файла.

Упаковка – процесс, напоминающий укладку чемодана, заключающийся в том, что путём изменения местоположения и взаимного расположения носителей информации удается сократить её. Это один из самых старых и по сей день популярных способов сжатия информации, о котором помнят, однако, только специалисты. Он отличается исключительной простотой и высокой скоростью работы. Вот лишь краткий перечень задач, элегантно решаемых с помощью упаковки:

Многие знают, что файлы на дисках хранятся по секторам. Размер сектора обычно равен или кратен 512 байтам, размеры же файлов самые разные. Поскольку кластер является минимальной, неделимой единицей объёма диска и состоит из нескольких секторов, то файлы длиной 2 и 508 байт займут один

кластер, а файлы размером в 513 и 1001 байт займут два. В любом случае теряется до 511 байт. Программы, способные объединять вместе несколько коротких файлов в один большой, дают ощутимое сокращение занимаемого ими объёма.

Коды переменной длины основаны на том, что в тексте не все буквы имеют одинаковую степень встречаемости. Например, гласные буквы встречаются гораздо чаще шипящих, а тем более чаще букв Ь, Ъ, Ё, Й. Поэтому все буквы алфавита можно расположить по степени их встречаемости и кодировать различным количеством бит. Те буквы, которые встречаются наиболее часто, необходимо кодировать меньшим количеством бит, а те, которые встречаются редко, можно кодировать большим количеством бит. В результате записанный таким способом текст будет иметь меньший размер. Ярким примером метода, использующего коды переменной длины, является азбука Морзе.

Метод сжатия на основе словаря. Этот метод основан на том, что для сжимаемого текста сначала создаётся словарь строк. Под строкой понимается любая непрерывная последовательность символов (букв), которой в словаре соответствует указатель. Длина строки выбирается из диапазона 16 – 256 байт. В результате весь текст состоит из последовательности указателей, которые определяют последовательность строк в тексте. Самой сложной задачей в этом методе является процедура составления словаря. Для оптимизации этого процесса разработано несколько алгоритмов, изучение которых выходит за рамки данного повествования. Для ещё большего сжатия информации применяют двухступенчатые архиваторы; сжатый текст на основе метода со словарём подвергается ещё сжатию по одному из выше описанных методов. Подобную процедуру используют архиваторы ARJ, RAR, PKZIP.

В любом случае текст подвергается не очень сильному сжатию, в среднем до 4-6 раз. Хотя есть разновидности архиваторов, которые сжимают текст до 10 раз. Здесь следует заметить, что степень сжатия зависит не только от архиватора, использующего соответствующий метод, но в большей степени от самого текста. Чем больше в нём одинаковых выражений, повторяющихся фраз, тем сильнее он может быть сжат. Большинство архиваторов, обладающих высокой степенью сжатия, для разархивирования текста требуют получения всего архива, прежде чем они могут приступить к извлечению текста из архива. Поэтому такие архиваторы не подходят для тех случаев, когда необходимо производить процесс разархивации в темпе поступления информации.

Графическая информация

Графическая информация лучше всего поддаётся сжатию. Это связано с тем, что в графической информации присутствует очень много почти одинако-

вых точек – пикселей. Для сжатия графической информации очень хорошо подходят методы, связанные с кодированием длинных последовательностей. Суть этих методов сводится к тому, что любой последовательности повторяющихся входных символов ставится в соответствие набор из трёх выходных символов: первый – префикс, говорящий о том, что встретилась входная повторяющаяся последовательность, второй – определяет длину повторяющейся последовательности, третий – сам входной символ.

Лучше всего работу метода пояснить на конкретном примере. Например, пусть имеется (шестнадцатеричный) текст вида

05 05 05 05 05 05 01 01 03 03 03 03 03 03 05 03 FF FF FF FF

состоящий из 20 байт. Допустим, в качестве префикса выбран символ – FF. Тогда на выходе архиватора мы получим последовательность

FF 06 05 FF 02 01 FF 06 03 FF 01 05 FF 01 03 FF 04 FF

Её длина – 18 байт, то есть достигнуто некоторое сжатие. Однако нетрудно заметить, что при кодировании некоторых символов размер выходного кода возрастает (например, вместо 01 01 – FF 02 01). Очевидно, одиночные или дважды (трижды) повторяющиеся символы кодировать не имеет смысла – их надо записывать в явном виде. С учётом всего вышесказанного получим новую последовательность:

FF 06 05 01 01 FF 06 03 05 03 FF 04 FF

длиной 13 байт. Достигнутая степень сжатия: 13/20 = 65%.

Нетрудно заметить, что префикс может совпасть с одним из входных символов. В этом случае входной символ может быть заменен своим «префиксным» представлением, например:

FF то же самое, что и FF 01 FF (три байта вместо одного)

Поэтому от правильного выбора префикса зависит качество самого алгоритма сжатия, так как, если бы в нашем исходном тексте часто встречались одиночные символы FF, размер выходного текста мог бы даже превысить входной. Некоторая разновидность описанного выше метода реализована для сжатия изображений в формате PCX. В общем случае в качестве префикса следует выбирать самый редкий символ входного алфавита. А теперь обратимся к изображению, представленному на рисунке 4.8. Этот рисунок хорошо поддаётся сжатию, поскольку имеет большое количество одинаковых пикселей.

В последнее время широкое распространение получила видеоинформация, которая представляет собой последовательность мало чем отличающихся друг от друга графических изображений, особенно при медленно меняющемся

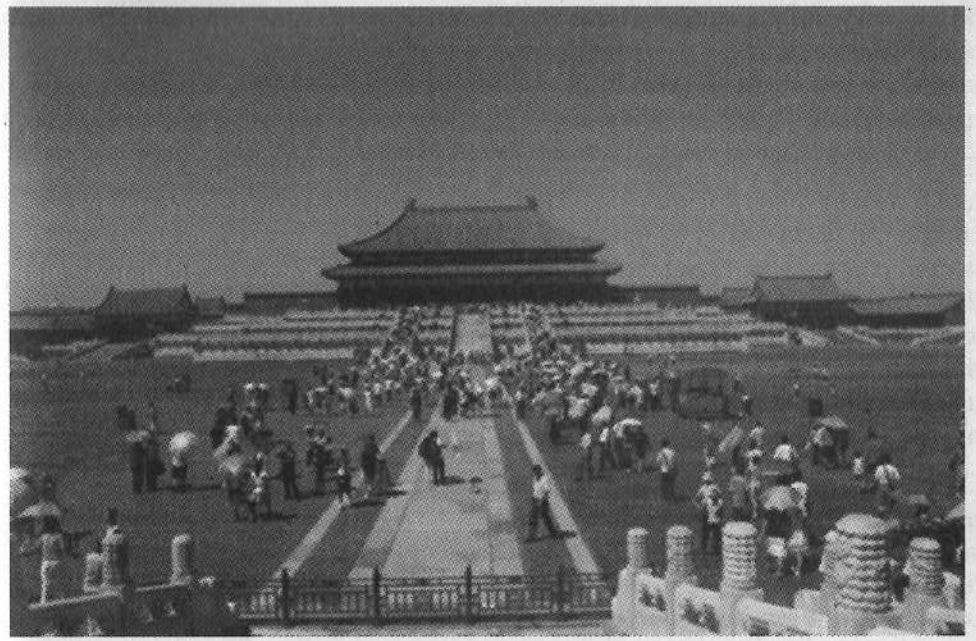


Рис. 4.8. Пример для сжатия

действии. Действительно, в кино частота изменения кадров составляет 24 Гц, а телевизионный формат предполагает смену кадров с частотой 25 Гц. Поэтому для сжатия видеоинформации можно использовать метод относительного изменения, который заключается в том, что начальное графическое изображение сжимается методом длинных последовательностей, а для последующих кадров достаточно кодировать только изменения относительно предыдущего. В результате видеоинформацию можно сжать в 100 и более раз без потери качества.

Достоинством методов кодирования видеоинформации является возможность их раскодирования в темпе поступления информации, что делает возможным их применение в цифровом телевидении.

Поскольку не все потребители видеоинформации обладают устройствами отображения высокого качества, то для них можно кодировать видеоинформацию с потерей качества, которая на аппаратуре низкого качества будет незаметна, вследствие того, что качество аппаратуры хуже, чем качество закодированного изображения.

Например, вы купили проигрыватель Blue Ray, DVD-диск с фильмом в формате высокой чёткости – HDTV и выводите всё это на телевизор с диагональю

54 см, который не поддерживает режим высокой чёткости. Естественно, ни о каком высоком качестве изображения не может быть и речи. С таким же успехом вы могли смотреть фильм с обычным качеством, не заметив разницы.

Сжатие информации с потерей качества открывает широкие возможности для разработчиков архиваторов. Ни один из существующих мониторов не в состоянии воспроизвести цветовую гамму из 16 млн. цветов, поэтому количество разрядов на кодирование цвета с 24 можно сократить, например, до 21, и это не скажется на качестве. В сотовых телефонах всего 65000 цветов, а качество отличное, поэтому можно сократить количество разрядов на кодирование цвета до 16. Специально для сжатия видеоинформации создан стандарт MP4, позволяющий сжимать видеоинформацию как без потери, так и с потерей качества.

Одним из возможных вариантов сжатия видеоинформации является уменьшение разрешения изображения. Стандарт HDTV работает с разрешением только 1920 на 1080 пикселей, если телевизор не поддерживает такое разрешение, то его можно уменьшить до 1024 на 800 пикселей. В результате видеоизображение с потерей качества можно сжать в 1000 и более раз. Поэтому в последнее время в продаже появилось очень много фильмов, записанных с потерей качества на обычный DVD-диск (до 10-12 фильмов на одностороннем однослойном диске).

При воспроизведении видеоинформации искажение или потеря отдельных бит, пикселей или даже больших групп пикселей (до целого кадра) практически не сказываются на качестве изображения.

К сожалению, многие компании, предоставляющие услуги цифрового телевидения, грешат тем, что предоставляют видеоинформацию не надлежащего качества, экономя на устройствах хранения информации.

Звуковая информация

Звуковая информация хуже всех видов информации поддаётся сжатию. Это связано со специфическими особенностями восприятия звука человеческим ухом. Впервые с проблемой передачи звука в цифровом виде столкнулись в телефонии. Для качественной передачи всех основных гармоник собеседников, передаваемых голосовых сообщений, в телефонии был выбран диапазон частот от 300 до 3400 Гц.

Для качественной передачи голоса в цифровом виде используется частота квантования амплитуды звуковых колебаний 8000 Гц. Это связано с тем, что в соответствии с теоремой Найквиста – Котельникова для качественной

передачи голоса достаточно выбрать частоту дискретизации, в два раза превышающую самую высокую гармонику непрерывного сигнала, то есть 2 x 3400 = 6800 Гц. Выбранная в действительности частота дискретизации 8000 Гц кратна степени двойки, что очень важно для цифровой техники, она обеспечивает некоторый запас качества. Для представления амплитуды одного замера обычно используется 7 или 8 бит. Соответственно, это дает 127 или 256 градаций звукового сигнала по амплитуде, что оказывается вполне достаточным для качественной передачи голоса.

Таким образом, для передачи одного голосового канала необходима пропускная способность 56 или 64 Кбит/с в зависимости от того, каким количеством бит представляется каждый замер. Стандартным является цифровой канал со скоростью передачи 64 Кбит/с, который также называется элементарным каналом цифровых телефонных сетей.

Передача непрерывного сигнала в дискретном виде требует от сетей жесткого соблюдения временного интервала 125 мкс (соответствующего частоте дискретизации 8 КГц) между соседними замерами, то есть синхронной передачи данных между узлами сети. При несоблюдении синхронности прибывающих замеров исходный сигнал восстанавливается неверно, что приводит к его искажению. Так, задержка в периоде следования замеров в 10 мс может привести к эффекту «эха». У большинства собеседников при этом создаётся иллюзия, наоборот, очень качественного канала связи с эхом. А сдвиги между замерами в 200 мс приводят к потере распознаваемости произносимых слов.

В то же время потеря одного замера при соблюдении синхронности между остальными замерами практически не сказывается на воспроизводимом звуке. Это происходит за счёт сглаживающих устройств в цифроаналоговых преобразователях, которые основаны на свойстве инерционности любого физического сигнала, амплитуда звуковых колебаний не может мгновенно измениться на большую величину.

Существуют и другие методы, позволяющие представить замеры голоса в более компактной форме, например, стандарт CCITT основан на нахождении разностей между последовательными замерами голоса, которые затем и передаются по сети. Если разность кодировать четырьмя битами, то голос передается со скоростью 32 Кбит/с. Более современный метод LPC делает замеры исходной функции более редко, но использует методы прогнозирования направления изменения амплитуды сигнала. При помощи этого метода можно понизить скорость передачи голоса до 9600 бит/с.

Представленные в цифровой форме непрерывные данные легко можно передать через цифровую сеть. Для этого достаточно поместить несколько замеров

в кадр какой-нибудь стандартной сетевой технологии, снабдить кадр правильным адресом назначения и отправить адресату. Адресат должен извлечь из кадра замеры и подать их с частотой квантования (для голоса – с частотой 8000 Гц) на цифроаналоговый преобразователь. По мере поступления следующих кадров с замерами голоса операция должна повториться. Если кадры будут прибывать достаточно синхронно, то качество голоса может быть достаточно высоким.

Однако кадры в компьютерных сетях могут задерживаться как в конечных узлах (при ожидании доступа к разделяемой среде), так и в промежуточных коммуникационных устройствах. Поэтому качество голоса при передаче в цифровой форме через компьютерные сети обычно бывает невысоким. Для повышения качества голоса в последнее время в конечных узлах стали устанавливать буферы. Процесс воспроизведения начинается только после заполнения буфера. Если буфер имеет достаточную ёмкость, то неравномерность в поступлении кадров может быть практически сведена к нулю.

Для кодирования музыкальной фонограммы в начале 80-х годов двумя фирмами Sony и Philips был разработан метод записи стереофонической (2-х канальной) музыкальной информации на CD – compact disk в соответствии со стандартом самого высокого качества Hi-Fi (High Fidelity – высокая верность воспроизведения), и устройство записи и воспроизведения. Этот метод обеспечивает настолько высокое качество записи, что только в 21 веке удалось создать аппаратуру, обеспечивающую предложенное качество за баснословную цену.

Цифровой стандарт записи оговаривает следующие основные характеристики: диапазон частот 20 – 20000 Гц, частоту дискретизации – 44,1 КГц, точность представления информации – 16 бит, длительность записи на CD – до 70 минут (в 2-х канальном варианте), максимальное количество информации на диске – 680 Мбайт.

Слово СТЕРЕО в переводе с греческого языка означает объёмное, а не 2-х канальное воспроизведение. Развитию многоканальных систем способствовало снижение цен на радиоаппаратуру. После многочисленных исследований оказалось, что 2-х канальная звуковая аппаратура даёт слушателю возможность ощущать пространственное звучание по фронту. То есть слушатель может различить местоположение отдельных инструментов на фронте (на сцене перед ним), при этом даже ухудшение качества воспроизведения до 20%, по сравнению с МОНО (одноканальной) системой, слушатель не замечает. Это связано с психическими особенностями человека.

Увеличение количества каналов до 4-х позволяет ощутить пространственную картину звучания в объёме. Оказалось, что такая 4-х канальная

(квадро) система, по сравнению с 2-х канальной, несомненно, обладает более высоким качеством, но на концертах ансамбль, певец или оркестр всегда располагаются перед слушателем, а не вокруг него, поэтому ощутимого выигрыша в качестве звучания такая система, при двукратном увеличении стоимости, не даёт. Кроме этого, "ложку дёгтя" в квадросистемы внесли звукорежиссёры, которые пытались "посадить" слушателя прямо в середину оркестра или на сцену. В результате квадросистемы не прижились.

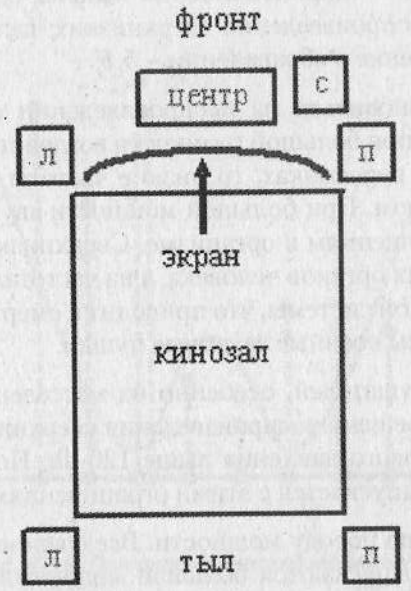


Рис. 4.9. Система Dolby surround

Другое дело кинотеатр, огромный экран, тёмный зал, на котором происходит завораживающее действие, хороший звук, зрителя переполняют эмоции, а если ещё добавить несколько звуковых спецэффектов типа перемещающихся в пространстве звуков сзади, слева и справа, то зрители получат неизгладимые впечатления. Поэтому для кинотеатров многоканальная система воспроизведения звука себя оправдывает, поскольку привносит зрителю ощущение нахождения прямо в гуще событий, происходящих на экране.

Специально для кинотеатров лабораторией Dolby была разработана система окружающего звучания Dolby surround. Система окружающего звучания Dolby surround определяет наличие 4-х широкополосных каналов – двух фронтальных и двух тыловых, диапазон воспроизводимых частот которых от 200 до 20000 Гц. Воспроизведение диалогов и монологов обеспечивает узкополосный (диапазон воспроизводимых частот от 200 до 10000 Гц)

центральный канал, расположенный непосредственно за экраном. Поскольку стереоэффект не наблюдается в диапазоне низких частот (ниже 200 Гц), а воспроизведение низких частот (ниже 100 Гц) сопряжено с большими техническими трудностями, то для снижения стоимости всех акустических систем решено было использовать один, чисто низкочастотный канал, воспроизводящий частоты ниже 200 Гц, который получил название subwoofer – репродуктор сверхнизкого тона. В результате система состоит из шести каналов, четыре из которых – широкополосные каналы, один – узкополосный канал и один канал воспроизведения сверхнизких частот. Это обстоятельство нашло своё отражение в обозначении – 5.1.

Следует немного остановиться на воспроизведении сверхнизких частот. Если высокие частоты при большой громкости воздействуют на уши, вплоть до боли в барабанных перепонках, то низкие частоты человек в основном воспринимает всем телом. При большой мощности звука это может привести к неприятным ощущениям в организме. Сверхнизкие частоты вызывают резонанс внутренних органов человека, а на частотах в районе 7-8 Гц наступает резонанс нервной системы, что приводит к смертельному исходу. На этом принципе устроены военные звуковые пушки.

Чтобы не калечить слушателей, особенно на массовых мероприятиях, во всех странах мира запрещено воспроизведение сверхнизких частот – ниже 60 Гц и создание звукового давления выше 120 db. Поэтому вся кинотеатральная аппаратура выпускается с этими ограничениями.

И ещё одно замечание по поводу мощности. Все современные звуковоспроизводящие аппараты выпускаются большой мощности. Например, какой-нибудь маленький переносной аппарат, весом 3-4 килограмма, имеет на своём корпусе рекламную надпись типа "600 ватт". Это совсем не значит, что он действительно будет громко играть, поскольку киловаттная конфорка электропечи вообще не издаёт ни единого звука. Это говорит о том, что она плохо преобразовывает электрические токи в звуковые колебания. Поэтому мощность звуковоспроизводящей аппаратуры вообще ни о чём не говорит.

Для отображения возможностей акустических систем в настоящее время принят один параметр – мощность необходимая для достижения звукового давления в 95 db на расстоянии в один метр. Поэтому акустические системы мощностью 250 Ватт и 50 Ватт будут создавать одинаковое звуковое давление (звучать одинаково громко).

Записанная для всех шести каналов звуковая информация имеет объём в три раза больше, чем двухканальная запись (при длительности фильма в 70 минут шестиканальный звук занимает около 2 Гбайт). Для сокращения количества звуковой информации применяют методы сжатия с потерей качества,

основанные на психоакустической модели уха, показанной на рис. 4.10.

Психоакустическая модель человеческого уха показывает чувствительность уха к разным частотам. Наивысшая чувствительность наблюдается в районе 1000 и 5000 Гц. Именно в районе этих частот создаются всевозможные тревожные сигналы, чтобы при минимальной мощности излучателя обеспечить наивысшую громкость.

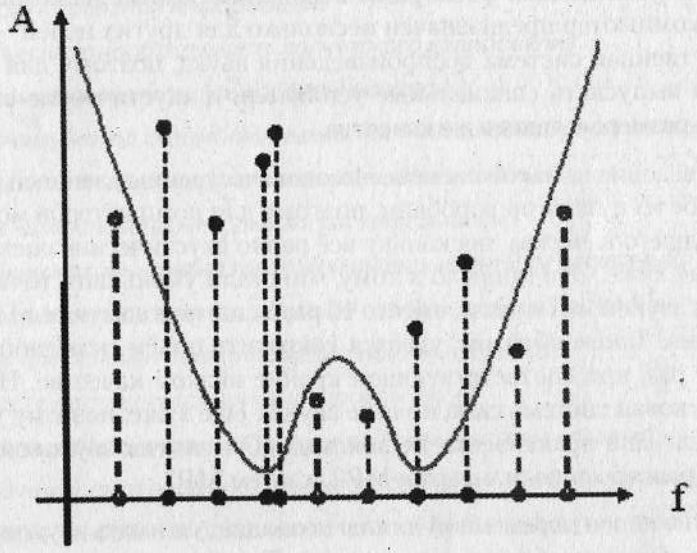


Рис. 4.10. Психоакустическая модель уха

Звуковой сигнал подвергается обработке – разложению на составляющие частоты. Для этого 1024, 2048 или 4096 последовательных измерений амплитуды звукового сигнала одного канала преобразуются в ряд составляющих частот посредством преобразования Фурье. Те частоты, амплитуда которых находится ниже порога слышимости (под кривой графика) удаляются из сигнала, их всё равно человеческое ухо не слышит. Для близко расположенных частот наблюдается эффект маскирования, частота с большей амплитудой заглушает частоту с меньшей амплитудой на некоторое время; поэтому меньшую амплитуду можно удалить из сигнала с учётом временного фактора. В результате информации становится меньше. При воспроизведении необходимо обратное преобразование, а оно требует достаточно большого количества вычислений.

В первых системах Dolby surround все преобразования выполнялись аналоговыми устройствами. С ростом вычислительных возможностей микроэлектроники появилась возможность производить преобразования в цифровом

виде по приемлемой цене, поэтому система, работающая с цифровой информацией, получила название Dolby digital surround. Система Dolby позволяет сжать звуковую информацию до 4-х раз при относительно высоком качестве.

С распространением CD-приводов в персональных компьютерах и появлением "приличных" звуковых преобразователей стало возможным воспроизведение музыкальных фонограмм в фоновом режиме на компьютере. Но всё-таки компьютер предназначен несколько для других целей – это не высококачественная система воспроизведения звука, поэтому для компьютеров стали выпускать специальные усилители и акустические системы небольшого размера и такого же качества.

Воспроизведение на такой системе высококачественных записей равносильно стрельбе из пушки по воробьям, поэтому для компьютеров можно иметь запись худшего качества, поскольку всё равно акустические системы и усилитель ещё хуже. Это привело к тому, что стали уменьшать точность представления звукового сигнала, вместо 16 разрядов используются 14, 12 и даже 10 разрядов. Таким образом, удаётся сократить объём исходной информации до 10 раз, при соответствующем крайне низком качестве. Но компьютерная звуковая система сама по себе звучит ещё хуже, поэтому ухудшение качества сигнала практически не заметно. Для сжатия звуковой информации были разработаны стандарты MP2, а затем MP3.

Применение кодов переменной длины позволяет сжимать звуковую информацию до 2,5 – 3 раз без потери качества. Такой метод использует система цифрового театра – DTS (Digital theater system).

Естественно, что для создания высококачественного объёмного звукового поля необходимо иметь бесконечное количество акустических систем по всему периметру помещения, где происходит прослушивание фонограмм. При этом стоимость такой системы и количество воспроизводимой информации тоже будут стремиться к бесконечности. Поскольку усилительная техника из года в год дешевеет, благодаря бурному развитию микроэлектроники, то возможно создание систем с большим количеством каналов, чем 5.1, при относительно невысокой стоимости. Поэтому стали появляться системы 7.1, 9.1, 11.1 и более, этому процессу в большей степени способствует появление высокоёмких носителей информации (CD-DVD, CD-Blue Ray).

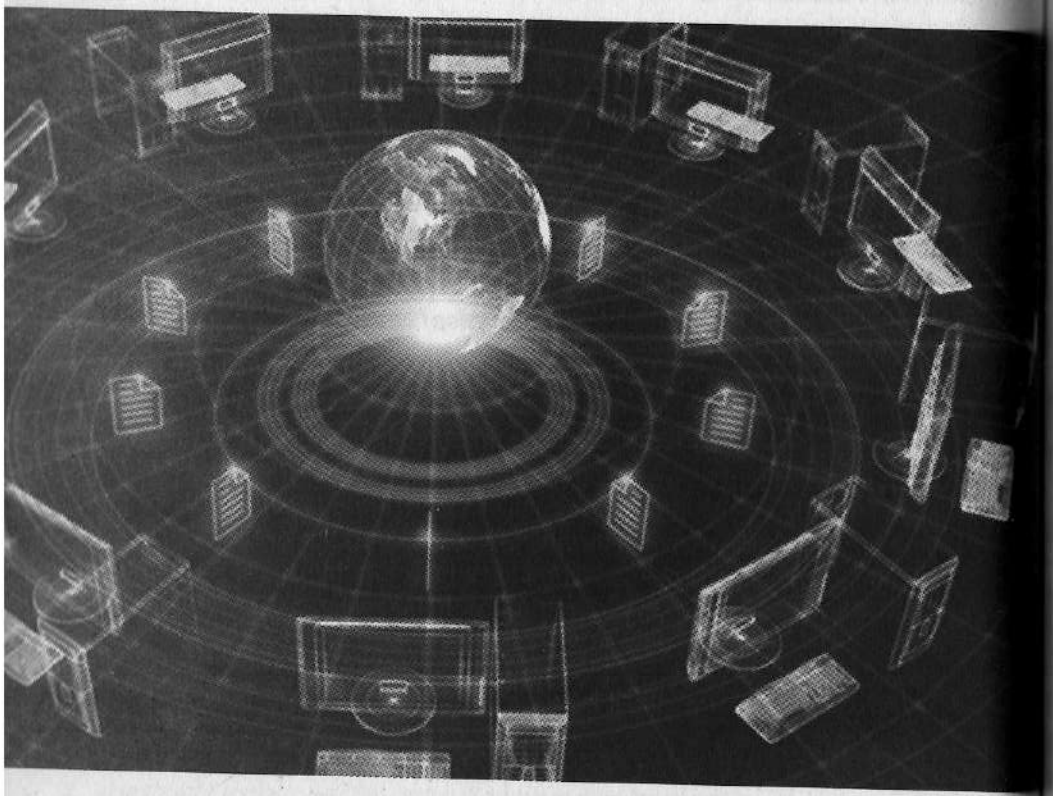
Контрольные вопросы

1. *В чём заключается суть метода дискретизации?*
2. *Назовите известные Вам используемые в сетях коды.*

3. *Почему потенциальный код не получил распространения в сетях?*
4. *Объясните причину появления такого разнообразия кодов.*
5. *Почему биполярный импульсный код требует более скоростного канала связи?*
6. *Назовите достоинства и недостатки манчестерского кода.*
7. *Каким образом можно с помощью одного импульса передать более одного бита полезной информации?*
8. *Объясните необходимость логического кодирования.*
9. *В чём заключается суть избыточных кодов?*
10. *Почему метод скремблирования получил широкое распространение?*
11. *Какие методы обнаружения ошибок Вам известны?*
12. *На чём основаны методы сжатия информации?*
13. *Какие методы сжатия текстовой информации Вам известны?*
14. *Расскажите суть методов сжатия графической информации.*
15. *На чём основан принцип сжатия видеоинформации?*
16. *Как представляется звук в цифровом виде?*
17. *Какими параметрами обладает высококачественный звук?*
18. *Почему для передачи голоса необходима скорость передачи 64 Кбит/с?*
19. *Какие требования предъявляются к каналам передачи голоса?*
20. *Почему не прижились четырёхканальные домашние системы воспроизведения звука?*
21. *Почему система Dolby surround имеет обозначение 5.1?*
22. *На чем основаны методы компрессии музыкальных данных?*
23. *Во сколько раз можно сжать текстовую, графическую и видеоинформацию?*
24. *Назовите максимальную степень сжатия звука при высоком качестве.*
25. *Какая система позволяет сжимать звук без потери качества?*

Глава 5.

АРХИТЕКТУРА ETHERNET



Термином Ethernet обозначают семейство продуктов для локальных сетей, описываемых стандартом IEEE 802.3, который определяет протокол, известный под названием CSMA/CD (множественный доступ с контролем несущей и обнаружением конфликтов). Первый вариант Ethernet был разработан корпорацией Xerox в 1970-х годах. Это была экспериментальная сеть, построенная на основе коаксиального кабеля, которая передавала данные со скоростью 3 Мбит/с. Разработка предназначалась для локальных сетей со случайным, но иногда весьма интенсивным обменом. Успех проекта сразу привлёк к нему внимание, и в 1980 году консорциум трёх компаний – DEC, Intel и Xerox – разработал спецификацию Ethernet 1.0 для передачи данных со скоростью 10 Мбит/с.

Первый стандарт IEEE 802.3 был основан на спецификации Ethernet 1.0 и очень был на неё похож. Проект стандарта был одобрен рабочей группой 802.3 в 1983 году, а в 1985 году опубликован как официальный стандарт (ASNI/1EEE Std. 802.3–1985). С тех пор был принят ряд дополнений к стандарту, отражающих новые достижения в технологиях и поддерживающих дополнительные сетевые среды и более высокие скорости передачи, а также несколько новых дополнительных возможностей управления доступом к сети.

В настоящее время существуют пять стандартных скоростей передачи данных по коаксиальному кабелю, витой паре и волоконно-оптическому кабелю:

- **10 Мбит/с – 10BaseT, 10Base5 и 10Base2**
- **100 Мбит/с – Fast Ethernet**
- **1000 Мбит/с – Gigabit Ethernet**
- **10000 Мбит/с – 10Gigabit Ethernet**
- **40000 Мбит/с – 40Gigabit Ethernet**

В острой конкурентной борьбе Ethernet выжил и стал основной технологией локальных сетей (в настоящее время она применяется примерно для 85% персональных компьютеров и рабочих станций, подключенных к локальной сети), так как её протокол обладает следующими свойствами:

- простота, легкость реализации, управления и обслуживания;
- дешевизна сетей;
- значительная топологическая гибкость при установке сетей;
- гарантия успешного взаимодействия между продуктами совместимых стандартов независимо от производителя.

В качестве метода доступа в сетях Ethernet используется метод CSMA/CD, который применяется исключительно в сетях с логической общей шиной. Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно получать данные, которые любой из компьютеров начал передавать на общую шину. Все данные, передаваемые по сети, помещаются в кадры определённой структуры и снабжаются уникальным адресом станции назначения. Кадр данных содержит семь полей (рис. 5.1):

- **PRE** – заголовок. Длина – 7 байт. Заголовок – это набор чередующихся единиц и нулей, по которому принимающие станции узнают о поступающем кадре. Он также служит для синхронизации приёмников.
- **SFD** – признак начала кадра. Длина – 1 байт. Признак начала кадра – это набор чередующихся единиц и нулей, заканчивающийся двумя единицами, который означает, что следующий бит является началом адреса приёмника.
- **DA** – адрес приёмника. Длина – 6 байт. В поле адреса приёмника указывается, какой узел (узлы) должны получить этот кадр. Левый бит в поле адреса приёмника показывает, является ли адрес индивидуальным (0) или групповым (1), а следующий бит – является ли адрес глобально (0) или локально (1) управляемым. Оставшиеся 46 битов заполняются уникальным значением, которое идентифицирует станцию, группу станций или все станции в сети.
- **SA** – адрес источника. Длина – 6 байт. Адрес источника идентифицирует посылающий узел. Адрес источника – это всегда индивидуальный адрес, и левый бит в поле адреса источника всегда равен 0.
- **Длина/тип**. Длина поля – 4 байта. В этом поле указывается количество байт данных, содержащихся в поле данных кадра, или идентификатор типа кадра, если кадр собран по дополнительному формату. Если значение этого поля меньше или равно 1500, то количество байт в поле данных равно значению поля длина/тип. Если значение поля длина/тип больше 1536, то это кадр дополнительного типа, и значение поля длина/тип определяет тип кадра.
- **Данные**. Последовательность из n байт с любыми значениями, где n меньше или равно 1500. Если длина поля данных меньше 46, то его нужно дополнить до этого размера холостыми (нулевыми) байтами.
- **CRC** – контрольная сумма кадра. Длина – 4 байта. Эта последовательность содержит 32-разрядное значение остатка от деления содержимого кадра без учёта полей заголовка и признака начала кадра на образующий полином R(x) следующего вида:

$$R(x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$$

Проверка правильности принятого кадра осуществляется на аппаратном уровне в два этапа:

- на первом этапе проверяется корректность кадра по длине и кратности целому числу байт;
- на втором этапе производится деление содержимого кадра с учётом поля CRC на образующий полином. Если в остатке получился ноль, то кадр принят без ошибки. Вероятность пропуска ошибки при таком методе контроля составляет $\sim 2^{-32}$.

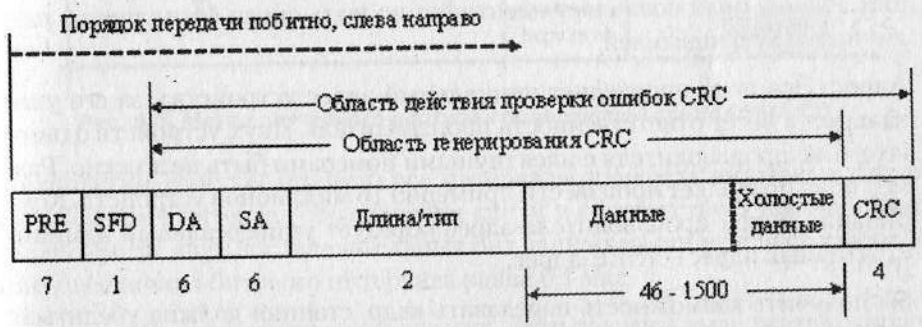


Рис. 5.1. Формат кадра данных в технологии Ethernet

Для адресации узлов в технологии Ethernet используются физические **MAC-адреса – адреса управления доступом к среде передачи (Media Access Control) канального уровня модели OSI. MAC-адрес уникален, представляет собой 48-разрядное двоичное число, содержащее следующие поля** (рис. 5.2):

- **И/Г** – индивидуальный/групповой. Этот признак определяет, к какому классу принадлежит адрес. Если признак И/Г=0, то адрес является индивидуальным адресом сетевого объекта. Если признак И/Г=1, то адрес является групповым, он позволяют обращаться сразу ко всем устройствам в пределах сети, в этом случае дальнейшее разбиение адреса на поля теряет смысл.
- **У/М** – признак универсального или местного управления. Этот признак определяет механизм генерации адреса сетевого узла. Если признак У/М=1, то адрес устройства сгенерирован не производителем и ответственность за уникальность лежит на администраторе сети или на пользователе. Значение И/Г=У/М=0 характерно для стандартных уникальных адресов, присваиваемых устройству его изготовителем.

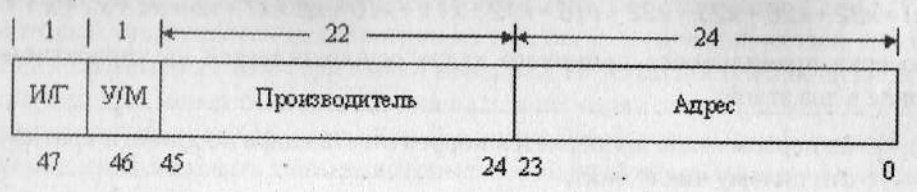


Рис. 5.2. Формат MAC-адреса

- **Производитель**. Это поле определяет производителя сетевого устройства. Каждому производителю присваивается одно или несколько значений. Размер поля позволяет идентифицировать около 4 миллионов различных производителей.
- **Адрес**. Это поле определяет уникальный адрес устройства, за его уникальность несёт ответственность производитель. Двух устройств одного и того же производителя с идентичными номерами быть не должно. Размер поля позволяет произвести примерно 16 миллионов устройств. Комбинация полей производитель-адрес образует универсальный административный адрес – IEEE-адрес.

Чтобы получить возможность передавать кадр, станция должна убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой. Признаком незанятости среды является отсутствие в ней несущей частоты, которая при манчестерском способе кодирования равна 5-10 МГц, в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

Если среда свободна, то узел имеет право начать передачу кадра. На рис. 5.3 приведён пример работы трёх узлов. Узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В сети Ethernet по общей шине сигналы передатчика узла 1 распространяются в обе стороны, так что их получают все узлы сети. Все узлы, подключённые к шине, могут распознать факт передачи кадра, и тот узел, который узнаёт собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, а затем, если требуется, посылает по шине кадр-ответ. Адрес узла-источника содержится в исходном кадре, поэтому узел-получатель знает, кому нужно послать ответ. Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаружил, что среда занята, на ней присутствует несущая частота, – поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

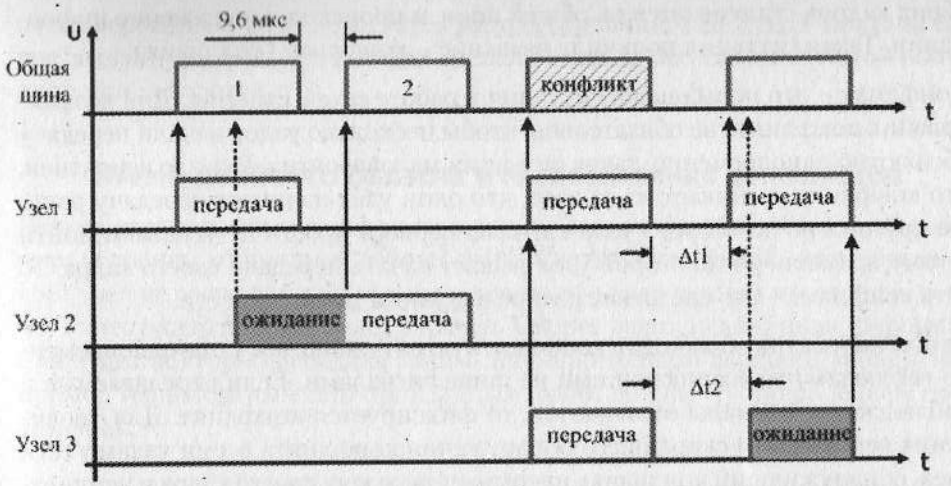


Рис. 5.3. Метод случайного доступа к передающей среде CSMA/CD

В технологии Ethernet принято все интервалы измерять в битовых интервалах; битовый интервал обозначается как bt и соответствует времени между появлением двух последовательных бит данных на шине; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу, равную 96 битовым интервалам, которая составляет 9,6 мкс. Эта пауза, называемая также межкадровым интервалом, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одним узлом.

После окончания технологической паузы узлы имеют право начать передачу своего кадра, поскольку среда свободна. Из-за задержек распространения сигнала по шине не все узлы одновременно фиксируют факт окончания передачи кадра узлом 1. Это связано с тем, что скорость света хоть и имеет большую величину (300000 км/с в вакууме), но всё же конечную. В приведённом примере (рис. 5.3) узел 2 дождался окончания передачи кадра узлом 1, выдержал паузу в 9,6 мкс и начал передачу своего кадра. При описанном подходе возможна ситуация, когда два узла одновременно пытаются передать кадр данных по общей среде.

Возникновение конфликтов

Механизм прослушивания среды и пауза между кадрами не гарантируют от возникновения такой ситуации, когда два или более узлов одновременно решают, что среда свободна, и они начинают передавать свои кадры, содержимое

обоих кадров сталкивается на общей шине и происходит искажение информации. Такая ситуация получила название – конфликт (коллизия).

Конфликт – это нормальная ситуация в работе сетей Ethernet. Для возникновения конфликта не обязательно, чтобы несколько узлов начали передачу абсолютно одновременно, такая ситуация маловероятна. Гораздо вероятней, что конфликт возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра. То есть конфликт – это следствие распределенного характера сети.

Чтобы корректно обработать конфликтную ситуацию, все узлы одновременно наблюдают за возникающими на шине сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется конфликт. Для увеличения вероятности скорейшего обнаружения конфликта всеми узлами сети узел, обнаруживший коллизию, прерывает передачу своего кадра и усиливает ситуацию конфликта посылкой в сеть специальной последовательности из 32 бит, называемой jam-последовательностью.

После этого обнаруживший конфликт передающий узел обязан прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем он может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему закону:

$$Пауза = L \times (интервал\ отсрочки)$$

где интервал отсрочки равен 512 битовым интервалам, а L представляет собой целое число, выбранное с равной вероятностью из диапазона $[0 - 2^N]$, где N – номер повторной попытки передачи данного кадра: 1,2, ..., 10.

После 10-й попытки интервал, из которого выбирается пауза, не увеличивается. Таким образом, случайная пауза может принимать значения от 0 до 52,4 мс. Если 16 последовательных попыток передачи кадра вызывают конфликт, то передатчик должен прекратить попытки передачи и отбросить этот кадр.

Из описания метода доступа видно, что он носит вероятностный характер, и вероятность успешного получения в своё распоряжение общей среды зависит от загруженности сети, то есть от интенсивности возникновения в узлах сети потребности в передаче кадров. При значительной интенсивности конфликтов полезная пропускная способность сети Ethernet резко падает, так как сеть почти постоянно занята повторными попытками передачи кадров.

Следует отметить, что метод доступа CSMA/CD вообще не гарантирует узлу, что она когда-либо сможет получить доступ к среде. Конечно, при небольшой загрузке сети вероятность такого события невелика, но при коэффициенте использования сети, приближающемся к 1, такое событие становится

очень вероятным. Этот недостаток метода случайного доступа – плата за его чрезвычайную простоту, которая сделала технологию Ethernet самой популярной и недорогой.

Время двойного оборота и распознавание конфликтов

Чёткое распознавание конфликтов всеми узлами сети является необходимым условием корректной работы сети Ethernet. Если какой-либо передающий узел не распознает конфликт и решит, что кадр данных им передан верно, то этот кадр данных будет утрачен. Скорее всего, искажённая информация будет повторно передана каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения.

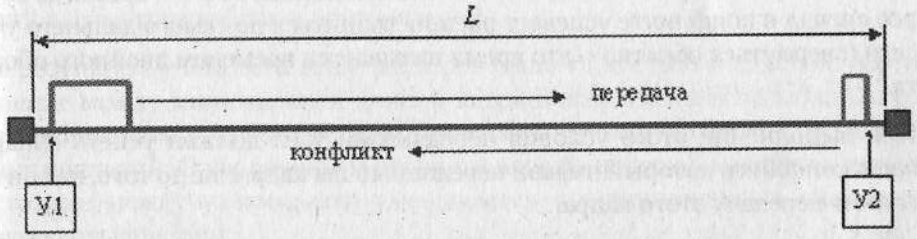


Рис. 5.4. Распознавание конфликта

Но повторная передача сообщения протоколами верхних уровней произойдет через значительно более длительный интервал времени (иногда даже через несколько секунд) по сравнению с микросекундными интервалами, которыми оперирует протокол Ethernet. Поэтому если конфликты не будут надёжно распознаваться узлами сети Ethernet, то это приведёт к заметному снижению полезной пропускной способности данной сети.

На надёжное распознавание конфликта непосредственное влияние оказывает скорость распространения электрического сигнала в передающей среде – скорость света. Для пояснения этого утверждения обратимся к рис. 5.4. Предположим, что на общей шине имеются всего два узла, расположенные на концах линии. Поскольку сигнал в линии распространяется в обоих направлениях, то отражённый сигнал от конца линии может исказить полезный сигнал. Для предотвращения отражения сигнала от конца линии необходимо на концах линии установить гасители сигнала.

Допустим, узел 1 решил передать пакет, он прослушал общую шину, убедился, что никто ничего не передаёт, и начал передачу. Электрические сигналы начали распространяться в оба конца. Рядом с узлом 1 установлен гаситель сигнала, поэтому отражения не происходит. Поскольку линия достаточно

длинная, сигналы до второго узла доходят только через некоторое время. А пока они не дошли, узел 2 решает тоже начать передачу, рядом с ним тоже расположен гаситель сигнала. Только узел 2 начинает передачу, как тут же происходит их наложение на сигнал от узла 1. Узел 2 фиксирует конфликт и прекращает передачу. Искажённые сигналы начинают распространяться в сторону первого узла. И если они придут после того, как узел 1 закончил передачу, то узел 1 не распознает возникшую конфликтную ситуацию и будет считать, что пакет передан успешно.

Для надежного распознавания конфликтов должно выполняться следующее соотношение:

$$t_{min} \geq \text{PDV}$$

где t_{min} – время передачи кадра минимальной длины, а *PDV* – время, за которое сигнал о конфликте успевает распространиться до самого дальнего узла сети и вернуться обратно – это время называется временем двойного оборота.

При выполнении этого условия передающий узел должен успеть обнаружить конфликт, который вызвал переданный им кадр, ещё до того, как он закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от длины минимального кадра и пропускной способнос ти сети, а с другой стороны, от длины общей шины сети и скорости распространения сигнала в среде передачи (для разных типов передающей среды эта скорость несколько отличается от скорости света).

В стандарте Ethernet принято, что минимальная длина поля данных кадра составляет 46 байт, что вместе со служебными полями и преамбулой даёт минимальную длину кадра – 72 байт или 576 бит. Отсюда может быть определено ограничение на расстояние между узлами.

Итак, в 10-мегабитном Ethernet время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время двойного оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от передающей среды и для толстого коаксиального кабеля равно примерно:

$$L = t \times C = 13280 \text{ м}$$

Учитывая, что за это время сигнал должен пройти по линии связи дважды, то расстояние между двумя узлами не должно быть больше 6 635 м. Для обеспечения хорошей распознаваемости сигнала на приёмном конце кабеля, при относительно маломощных передатчиках в узлах, максимальная длина

непрерывного сегмента толстого коаксиального кабеля, с учётом вносимого им затухания, выбрана в 500 м, а общий размер сети – 2500 м.

Повторители увеличивают мощность передаваемых с сегмента на сегмент сигналов, в результате затухание сигналов уменьшается и можно использовать сеть гораздо большей длины, состоящую из нескольких сегментов. В коаксиальных реализациях Ethernet разработчики ограничили максимальное количество сегментов в сети пятью, что в свою очередь ограничивает общую длину сети 2500 метрами. Даже в такой многосегментной сети условие обнаружения конфликтов по-прежнему выполняется с большим запасом. Однако в действительности временной запас существенно меньше, поскольку в многосегментных сетях сами повторители вносят в распространение сигнала дополнительную задержку в несколько десятков битовых интервалов. Естественно, небольшой запас был сделан также для компенсации отклонений параметров кабеля и повторителей.

В результате учёта всех этих факторов было тщательно подобрано соотношение между минимальной длиной кадра и максимально возможным расстоянием между узлами сети, которое обеспечивает надежное распознавание коллизий. С увеличением скорости передачи кадров максимальное расстояние между узлами сети уменьшается пропорционально увеличению скорости передачи.

В табл. 5.1 приведены значения основных параметров процедуры передачи кадра стандарта 802.3, которые не зависят от реализации физической среды. Важно отметить, что каждый вариант физической среды технологии Ethernet добавляет к этим ограничениям свои, часто более строгие ограничения, которые также должны выполняться.

Таблица 5.1. *Параметры уровня MAC Ethernet*

Параметры	Значения
Битовая скорость	10 Мбит/с
Интервал отсрочки	512 битовых интервалов
Межкадровый интервал	9,6 мкс
Максимальное число попыток передачи	16
Максимальное число возрастания диапазона паузы	10
Длина jam-последовательности	32 бита
Максимальная длина кадра (без преамбулы)	1518 байт
Минимальная длина кадра (без преамбулы)	64 байт (512 бит)
Длина преамбулы	8 байт
Минимальная длина случайной паузы после конфликта	0 битовых интервалов
Максимальная длина случайной паузы после конфликта	524 000 битовых интервалов

Максимальная производительность сети Ethernet

Рассчитаем максимальную производительность сегмента Ethernet в таких единицах, как число переданных кадров (пакетов) минимальной длины в секунду. Для расчёта максимального количества кадров минимальной длины, проходящих по сегменту Ethernet, заметим, что размер кадра минимальной длины вместе с преамбулой составляет 72 байт или 576 бит (рис. 5.1), поэтому на его передачу затрачивается 57,5 мкс. Прибавив межкадровый интервал в 9,6 мкс, получаем, что период следования кадров минимальной длины составляет 67,1 мкс. Отсюда **максимально возможная пропускная способность сегмента Ethernet составляет:**

$$K_{кор} = \frac{1}{67{,}1} = 14880 \text{ пак/с}$$

Естественно, что наличие в сегменте нескольких узлов снижает эту величину за счёт ожидания доступа к среде, а также за счёт конфликтов, приводящих к необходимости повторной передачи кадров.

Кадры максимальной длины технологии Ethernet имеют поле длины 1500 байт, что вместе со служебной информацией даёт 1518 байт, а с преамбулой составляет 1526 байт или 12208 бит. **Максимально возможная пропускная способность сегмента Ethernet для кадров максимальной длины составляет:**

$$K_{кор} = \frac{1}{1230{,}1} = 813 \text{ пак/с}$$

Теперь рассчитаем, какой максимальной полезной пропускной способностью в бит/с обладают сегменты Ethernet при использовании кадров разного размера.

Под полезной пропускной способностью понимается скорость передачи пользовательских данных, которые переносятся полем данных кадра. Эта пропускная способность всегда меньше номинальной битовой скорости Ethernet за счёт нескольких факторов:

- служебной информации кадра;
- межкадровых интервалов;
- ожидания доступа к среде.

Для кадров минимальной длины полезная пропускная способность равна:

$$C_п = 14800 \times 46 \times 8 = 5{,}48 \text{ Мбит/с}$$

Это намного меньше 10 Мбит/с, но следует учесть, что кадры минимальной длины используются в основном для передачи квитанций, так что к передаче собственно данных файлов эта скорость отношения не имеет.

Для кадров максимальной длины полезная пропускная способность

$$Cп = 813 \times 1500 \times 8 = 9{,}76 \text{ Мбит/с}$$

что весьма близко к номинальной скорости протокола.

Такой скорости можно достигнуть только в том случае, когда двум взаимодействующим узлам в сети Ethernet другие узлы не мешают, что бывает крайне редко. При использовании кадров среднего размера с полем данных в 512 байт пропускная способность сети составит 9,29 Мбит/с, что тоже достаточно близко к предельной пропускной способности в 10 Мбит/с.

В стандарте Ethernet допускается использование нескольких типов кабелей: коаксиальный кабель, витая пара и оптический кабель.

5.1. Толстый Ethernet

Во время рождения Ethernet коаксиальный кабель был единственным высокоскоростным кабелем, обладающим хорошей помехозащищённостью, поэтому первые сети по технологии Ethernet были созданы на коаксиальном кабеле.

Ethernet на толстом коаксиальном кабеле в основном соответствует экспериментальной сети Ethernet фирмы Xerox и может считаться классическим, его описывает стандарт 10Base-5. Ethernet использует в качестве среды передачи данных коаксиальный кабель с волновым сопротивлением 93 Ом, диаметром центрального медного провода 2,17 мм и внешним диаметром 0,5 дюйма ~ 12,5 мм (поскольку кабель достаточно толстый, его так и назвали – толстый Ethernet).

Топология сети Ethernet на толстом коаксиальном кабеле приведена на рис. 5.5. По коаксиальному кабелю сигналы распространяются в виде волны. На концах кабеля установлены гасители сигнала, поглощающие распространяющиеся по кабелю сигналы и препятствующие возникновению отражённой волны, с волновым сопротивлением 93 Ом (в английском языке они называются – terminator). Заглушка представляет собой угольный стакан с конусом, в который заходит волна и постепенно гаснет (рис. 5.6).

Коаксиальный кабель, ограниченный с двух сторон гасителями (заглушками), называется сегментом, он должен быть обязательно заземлён, причём только с одной стороны. Это связано с тем, что Земля в разных точках имеет разный потенциал, поэтому заземление с двух сторон может привести к протеканию тока в изолирующей оплётке, что неминуемо приведёт к наведению вторичных токов в центральной жиле кабеля и тем самым к искажению сигнала.

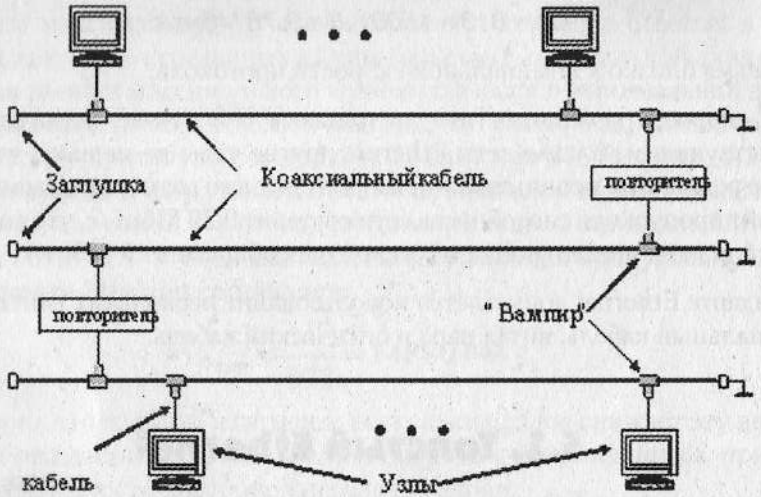


Рис. 5.5. Топология толстого Ethernet

Толстый Ethernet предназначался для соединения в сеть малых вычислительных машин, персональных машин в то время ещё не было, поэтому основная магистраль – коаксиальный кабель, который плохо гнётся и с ним тяжело работать, предполагалось прокладывать в специальных нишах в стенах помещений. Узлы подключаются к кабелю при помощи приёмопередатчика – трансивера, который устанавливается непосредственно на кабеле и питается от сетевого адаптера компьютера.

Трансивер может подсоединяться к кабелю как методом прокалывания, обеспечивающим непосредственный физический контакт, так и бесконтактным методом. Трансивер, использующий метод прокалывания, имеет и другое жаргонное название – «вампир» (рис. 5.7). Он соединяется с сетевым узлом трансиверным кабелем длиной до 50 м, состоящим из 15 проводников.

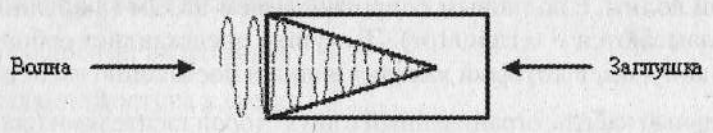


Рис. 5.6. Заглушка

К одному сегменту допускается подключение не более 100 трансиверов, причём расстояние между подключениями трансиверов должно быть от 2,5 м с шагом 0,5 м. На кабеле имеется разметка через каждые 0,5 м, которая обозначает точки подключения трансиверов. При подсоединении компьютеров в соответствии с разметкой влияние стоячих волн в кабеле на сетевые адаптеры сводится к минимуму.

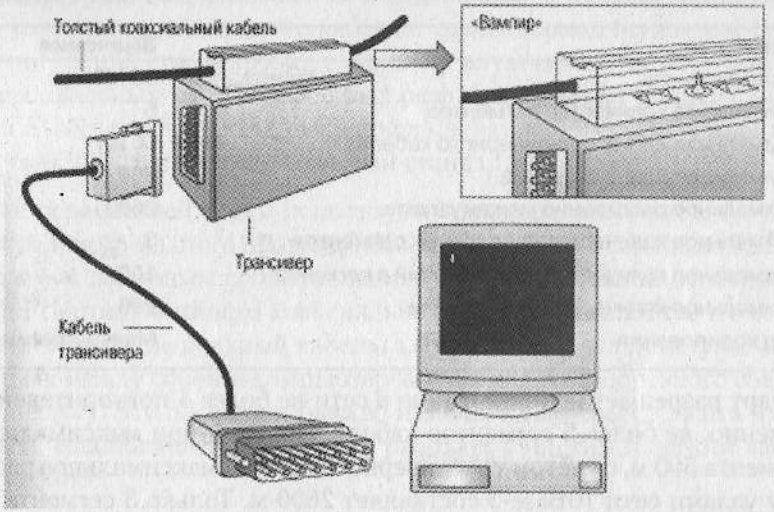


Рис. 5.7. Подключение узла к сети трансиверным кабелем

При возникновении неисправностей в адаптере может возникнуть ситуация, когда на кабель будет непрерывно выдаваться последовательность случайных сигналов. Так как кабель – это общая среда для всех узлов сети, то работа сети будет заблокирована одним неисправным адаптером. Чтобы этого не случилось, на выходе передатчика ставится схема, которая проверяет время передачи кадра. Если максимально возможное время передачи пакета превышается (с некоторым запасом), то эта схема просто отсоединяет выход передатчика от кабеля. Максимальное время передачи кадра (вместе с преамбулой) равно 1221 мкс, а время контроля устанавливается равным 4000 мкс.

Для объединения в одну сеть нескольких сегментов кабеля и увеличения тем самым общей длины сети служит повторитель, который принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Повторитель состоит из двух трансиверов, которые присоединяются к сегментам кабеля, а также блока повторения со своим тактовым генератором. Для лучшей синхронизации передаваемых бит повторитель задерживает передачу нескольких первых бит преамбулы кадра, за счёт чего увеличивается задержка передачи кадра с сегмента на сегмент, а также несколько уменьшается межкадровый интервал.

Таблица 5.2. *Основные характеристики сети Ethernet на толстом коаксиальном кабеле*

Параметры	Значения
Битовая скорость	10 Мбит/с
Максимальное количество сегментов	5
Максимальная длина трансиверного кабеля	50 м
Максимальная длина сегмента	500 м
Максимальное расстояние между узлами	2600 м
Максимальное количество населённых сегментов	3
Максимальное количество подключений в сегменте	100
Максимальное количество узлов в сети	296
Метод кодирования	Манчестерский код

Стандарт разрешает использование в сети не более 4 повторителей и, соответственно, не более 5 сегментов кабеля. Поэтому при максимальной длине сегмента 500 м, с учётом трансиверного кабеля, максимальное расстояние между узлами сети 10Base-5 составляет 2600 м. Только 3 сегмента из 5 могут быть нагруженными (населёнными), то есть такими, к которым подключаются конечные узлы. Между населёнными сегментами должны быть ненаселённые сегменты, так что максимальная конфигурация сети представляет собой два населённых крайних сегмента, которые соединяются ненаселёнными сегментами и ещё с одним населённым сегментом.

Правило применения повторителей в сети Ethernet 10Base-5 носит название «правило 5 – 4 – 3»: 5 сегментов, 4 повторителя, 3 населённых сегмента. Ограниченное число повторителей объясняется дополнительными задержками распространения сигнала, которые они вносят. Применение повторителей увеличивает время двойного оборота сигнала, которое для надёжного распознавания конфликтов не должно превышать время передачи кадра минимальной длины.

Каждый повторитель подключается к сегменту одним своим трансивером, поэтому к населённым сегментам можно подключить не более 99 узлов. Максимальное число конечных узлов в сети 10Base-5, таким образом, составляет 296 узлов. Основные характеристики сети Ethernet на толстом коаксиальном кабеле приведены в табл. 5.2.

5.2. Тонкий Ethernet

Появление персональных компьютеров привело к резкому увеличению интереса к сетям, но поскольку производимые для первых PC жёсткий диск и печатающее устройство стоили дороже самого компьютера, то для

эффективного использования таких дорогостоящих ресурсов сеть подходила как нельзя лучше. Использование очень жёсткого и плохо гнущегося (не гибкого) толстого коаксиального кабеля с трансиверами было неэффективным с точки зрения стоимости и условий эксплуатации. Поэтому специально для персональных компьютеров был разработан «облегчённый» вариант стандарта Ethernet с более тонким коаксиальным кабелем, получивший название – тонкий Ethernet, описываемый стандартом 10Base-2.

В качестве передающей среды используется коаксиальный кабель с диаметром центрального медного провода 0,89 мм и внешним диаметром 0,25 дюйма ~ 6,25 мм, с волновым сопротивлением 50 Ом. Основной особенностью стандарта Ethernet на тонком коаксиальном кабеле является то, что сегмент представляет собой не цельный кабель, а состоит из отдельных фрагментов, соединённых между собой и компьютером с помощью Т-образного соединителя (connector). Это позволяет быстро изменять размеры сегмента и модифицировать топологию сети, при этом работать с мягким и лёгким кабелем намного удобнее.

Поскольку сегмент состоит из фрагментов коаксиального кабеля, соединенных между собой с помощью разъёмов, то потери сигнала на каждом соединении приводят к ограничению максимальной длины сегмента на уровне 185 м. Сегмент должен иметь на концах гасители с волновым сопротивлением 50 Ом.

Максимальное количество узлов, подключаемых к одному сегменту, не должно превышать 30. Минимальное расстояние между станциями 1 м с шагом 0,5 м. На кабеле имеется разметка через каждые 0,5 м. Количество повторителей должно подчиняться правилу "5-4-3", тогда сеть будет иметь максимальную длину 5×185 = 925 м. Топология сети Ethernet на тонком коаксиальном кабеле приведена на рис. 5.8.

Так как Т-образный разъём непосредственно подключается к выходному

Рис. 5.8. Топология тонкого Ethernet

разъему платы сетевого адаптера, установленной в корпус компьютера, то кабель просто «висит» на сетевом адаптере, что может несколько затруднить физическое перемещение компьютеров при малой длине фрагмента.

Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, но он и наиболее сильно подвержен авариям и сбоям. Поскольку в канале передачи данных имеется большое количество механических соединений – каждый Т-образный разъём даёт три механических соединения, два из которых имеют жизненно важное значение для всей сети.

Таблица 5.3. *Основные характеристики сети Ethernet на тонком коаксиальном кабеле*

Параметры	Значения
Битовая скорость	10 Мбит/с
Максимальное количество сегментов	5
Максимальная длина сегмента	185 м
Максимальное расстояние между узлами	925 м
Максимальное количество населённых сегментов	3
Максимальное количество подключений в сегменте	30
Максимальное количество узлов в сети	86
Метод кодирования	Манчестерский код

Пользователи имеют доступ к разъёмам и могут нарушить целостность канала. Кроме того, эстетика и эргономичность этого решения оставляют желать лучшего, так как от каждой станции отходят два довольно заметных провода, которые под столом часто образуют моток кабеля – запас, необходимый на случай даже небольшого перемещения рабочего места.

После очередной уборки, произведённой «высокопрофессиональной» уборщицей, сеть обычно перестаёт работать. Для восстановления работоспособности сети необходимо проверить (прощупать) все механические соединения. При достаточно разветвлённой сети на это может уйти достаточно много времени. Основные характеристики сети Ethernet на тонком коаксиальном кареле приведены в табл. 5.3.

5.3. Ethernet на витой паре

Возросшие возможности микропроцессоров и падение цен на персональные компьютеры привели к лавинообразному использованию PC и, как следствие, соединению их в сети. Недостатки, присущие Ethernet на тонком коаксиальном кабеле, – небольшое количество компьютеров в сети и частные отказы из-за нарушения механических соединений – привели к необходимости разработать стандарт, позволивший объединять большее количество узлов в сеть с меньшим количеством отказов.

Витая пара, представляющая собой два изолированных провода, перевитых между собой и помещённых в защитную оболочку, оказалась очень технологичной линией связи.

Идея приспособить этот популярный вид кабеля для построения локальных сетей оказалась очень плодотворной, так как многие здания уже были оснащены нужной кабельной системой. Оставалось разработать способ подключения сетевых адаптеров и прочего коммуникационного оборудования к витой паре таким образом, чтобы изменения в сетевых адаптерах и программном обеспечении сетевых операционных систем были бы минимальными по сравнению с сетями Ethernet на коаксиальном кабеле.

Новый стандарт 10Base-T был принят в 1991 году как дополнение к существующему набору стандартов Ethernet. В стандарте 10Base-T используется физическая топология – звезда, при сохранении логической топологии общая шина. Для объединения компьютеров с использованием топологии звезда необходимо иметь центральное устройство – концентратор, к которому индивидуальными кабелями подключаются все узлы сети. Топология сети на витой паре приведена на рис. 5.9.

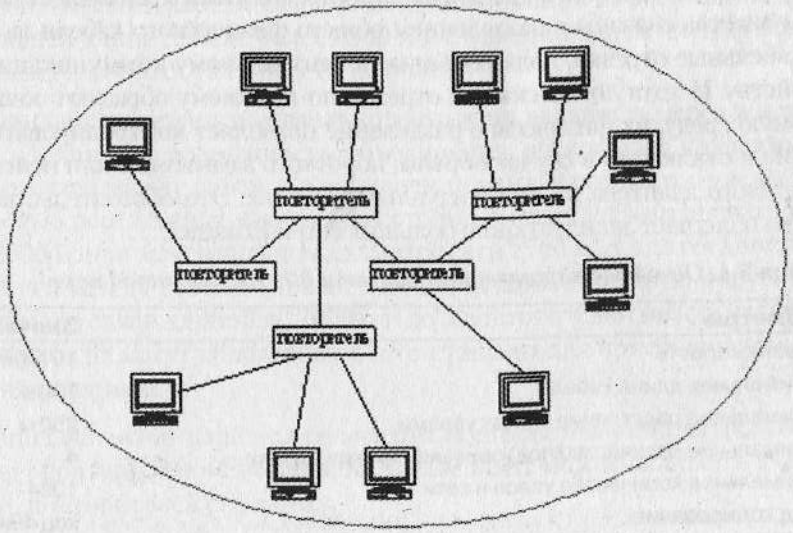


Рис. 5.9. Топология сети на витой паре

Концентратор (англоязычные термины – hub или concentrator) осуществляет функции повторителя сигналов на всех отрезках витых пар, подключенных к его портам, так что образуется единая среда передачи данных – логический канал (логическая общая шина). Стандарт определяет битовую

скорость передачи данных 10 Мбит/с и максимальное расстояние отрезка витой пары между двумя непосредственно связанными узлами (компьютерами и концентраторами) не более 100 м при наличии витой пары качества не ниже категории 3.

Концентраторы 10Base-T можно соединять друг с другом с помощью тех же портов, которые предназначены для подключения конечных узлов. При этом нужно позаботиться о том, чтобы передатчик и приемник одного порта были соединены соответственно с приемником и передатчиком другого порта.

Для обеспечения синхронизации узлов, при реализации процедур доступа CSMA/CD и надёжного распознавания узлами конфликтов, в стандарте определено максимальное число концентраторов между любыми двумя узлами сети, а именно 4. Это правило носит название «правила четырех повторителей» и заменяет правило «5-4-3», применяемое к коаксиальным сетям. При создании сети 10Base-T с большим числом узлов концентраторы можно соединять друг с другом иерархическим способом, образуя древовидную структуру.

Общее количество узлов в сети 10Base-T не должно превышать общего предела 1024. Сети, построенные на основе стандарта 10Base-T, обладают по сравнению с коаксиальными вариантами Ethernet многими преимуществами. Эти преимущества связаны с разделением общего физического кабеля на отдельные кабельные отрезки, подключенные к центральному коммуникационному устройству. И хотя логически эти отрезки по-прежнему образуют общую разделяемую среду, их физическое разделение позволяет контролировать их состояние и отключать в случае обрыва, короткого замыкания или неисправности сетевого адаптера на индивидуальной основе. Это обстоятельство существенно облегчает эксплуатацию больших сетей Ethernet.

Таблица 5.4. *Основные характеристики сети Ethernet на витой паре*

Параметры	Значения
Битовая скорость	10 Мбит/с
Максимальная длина кабеля	100 м
Максимальное расстояние между узлами	250 м
Максимальное количество повторителей между узлами	4
Максимальное количество узлов в сети	1024
Метод кодирования	код 4B/5B

Стандарт также оговаривает максимальное расстояние между любыми двумя узлами – оно не должно превышать 250 м. Поскольку вся сеть должна укладываться в окружность диаметром в 250 м, то этот параметр стали называть максимальным диаметром сети. Основные характеристики сети Ethernet на тонком коаксиальном кареле приведены в табл. 5.4.

Для построения сетей с удалёнными узлами данный стандарт не подходит, даже не в силу ограничений, связанных с витой парой, а из-за принципиальных ограничений метода CSMA/CD, рассчитанного на общую шину. Для того чтобы уйти от ограничения, накладываемого временем двойного оборота, был введён полнодуплексный (дуплексный) режим. В этом случае два узла обмениваются информацией по отдельным линиям, что исключает возможность занятости среды и как следствие – возникновение конфликтов (рис. 5.10), причём могут это делать одновременно. Витая пара для организации полнодуплексного режима подходит, поскольку имеет в своём составе 4 витых пары. Однако она позволяет организовать полнодуплексный режим на небольших расстояниях.

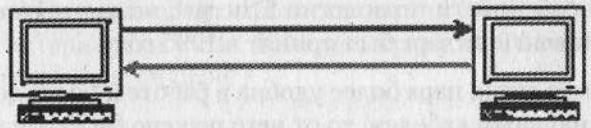


Рис. 5.10. Полнодуплексный режим

Для увеличения дальности передачи была разработана модификация стандарта 10Base-T, которая оговаривает возможность использования многомодового оптического кабеля для передачи информации на расстояние до 1000 м при общей длине сети не более 2500 м и с максимальным числом повторителей между любыми узлами сети не более 4-х.

Этот стандарт позволил значительно увеличить дальность передачи информации. Появилась возможность объединения нескольких удалённых локальных сетей между собой. Со временем появился стандарт с увеличенной мощностью передатчика, поэтому расстояние между узлами достигло величины 2000 м, при максимальном диаметре сети 2740 м. А для соединения повторителей между собой появилась модификация стандарта, допускающая иметь линию связи длиной до 2000 м и до 5 повторителей между узлами. Конечные узлы не могут использовать этот стандарт для присоединения к портам концентратора.

В Ethernet на витой паре под сегментом понимаются узлы, подключённые к одному повторителю. Естественно, что сегмент может включать в себя несколько повторителей с узлами.

5.4. Быстрый (Fast) Ethernet

10-мегабитный Ethernet устраивал большинство пользователей на протяжении около 15 лет. Однако в начале 90-х годов начала ощущаться его

недостаточная пропускная способность. Назрела необходимость в разработке "нового" Ethernet, то есть технологии, которая была бы такой же эффективной по соотношению цена/качество при производительности в 100 Мбит/с. В результате поисков и исследований специалисты разделились на два лагеря, что в конце концов привело к появлению двух новых технологий – Fast Ethernet и 100VG-AnyLAN. Они отличаются степенью преемственности с "классическим" Ethernet.

В 1992 году группа производителей сетевого оборудования, включая таких лидеров технологии Ethernet, как SynOptics, 3Com и ряд других, образовали некоммерческое объединение Fast Ethernet Alliance для разработки стандарта новой технологии, которая должна была в максимально возможной степени сохранить особенности технологии Ethernet, но иметь скорость передачи 100 Мбит/с. Новый стандарт был принят в 1995 году.

В связи с тем, что витая пара более удобна в работе и более дешёвая по сравнению с коаксиальным кабелем, то от него решено было отказаться вовсе. В качестве среды передачи оговаривались три типа кабеля: оптическое волокно, витая пара 3 категории и витая пара 5 категории. Для передачи информации по этим кабелям применяется потенциальный метод кодирования: 4B/5B для витой пары 5 категории и оптоволокна и 8B/6T для витой пары 3 категории. При идеологии общая шина применяется топология звезда.

Кодирование по методу 4B/5B заключается в том, что каждые 4 бита информации представляются 5 битами. Это необходимо для того, чтобы избавиться от длинных последовательностей нолей/единиц и повысить самосинхронизацию. Но за всё надо платить, поэтому битовая скорость полезной информации составляет не 100, а всего 80 Мбит/с. Оставшиеся 16 комбинаций решено было использовать для служебных целей. Они были разделены на три группы:

- 4 управляющие кодовые комбинации, передаваемые попарно и служащие признаками начала (SSD) и конца (ESD) кадра. Каждый кадр заключается между признаками начала и конца кадра. Первый байт префикса (см. рис. 5.1) заменяется кодовой парой SSD, а кодовая пара E5D добавляется в конце кадра после поля CRC;
- специальная кодовая комбинация IDLE, которая постоянно посылается в течение межкадровых интервалов для сохранения синхронизации между узлами. Получение сигнала IDLE интерпретируется как сообщение о том, что линия свободна;
- одиннадцать недействительных кодовых комбинаций, при получении любой такой кодовой комбинации весь входящий кадр рассматривается как ошибочный.

Для повышения производительности сети был стандартизован полнодуплексный режим. Поскольку в этом режиме узлы могут передавать информацию одновременно в обоих направлениях, используя только выделенную линию, то конфликтов не существует, что в значительной степени упрощает протокол доступа к среде. Небольшие модификации в заголовках пакетов были внесены для обеспечения совместимости полудуплексного и полнодуплексного Ethernet. Полнодуплексные связи чрезвычайно выгодны для соединения коммутаторов между собой.

Как было отмечено ранее, максимальное расстояние между узлами определяется временем двойного оборота, в связи с тем, что скорость передачи увеличилась в 10 раз, по сравнению с «классическим» Ethernet, то во столько же раз уменьшился и размер сети. Для стандарта Fast Ethernet он составляет 250 м. Все характеристики у Fast Ethernet, кроме скорости передачи, такие же, как у Ethernet на витой паре.

5.5. Скоростная технология Gigabit Ethernet

Достаточно быстро после появления на рынке продуктов Fast Ethernet сетевые интеграторы и администраторы почувствовали определенные ограничения при построении корпоративных сетей. Во многих случаях серверы, подключенные по 100-мегабитному каналу, перегружали магистрали сетей, работающие также на скорости 100 Мбит/с. Ощущалась потребность в следующем более скоростном стандарте. Окончательная версия стандарта Gigabit Ethernet была принята 1999 году.

При разработке стандарта Gigabit Ethernet перед разработчиками возникло несколько проблем. Во-первых, диаметр сети уменьшался до 25 м, во-вторых, кроме оптического кабеля, хотелось бы сохранить старое кабельное хозяйство на витой паре 5 категории и полнодуплексный режим передачи данных.

Проблема с диаметром сети была решена методом увеличения минимальной длины кадра с 64 байт (без учёта префикса) до 512 байт. Если кадр меньше минимального размера, то он увеличивается за счёт неинформативного поля, заполненного запрещёнными символами кода 8B/10B, которые невозможно принять за коды данных, при приёме кадра они удаляется (рис. 5.11).

Кроме того, для Gigabit Ethernet была добавлена пакетная передача кадров и применён метод кодирования 8B/10B. Пакетный режим – это функция, позволяющая посылать короткую последовательность кадров общей длиной до 8192 байт. Интервал между кадрами заполняется битами расширения – IFG, чтобы другие узлы сети видели, что сеть занята, и не пытались передавать

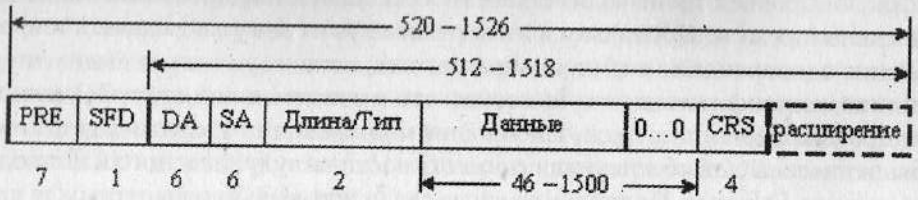


Рис. 5.11. Формат кадра Gigabit Ethernet

данные, пока не закончиться пакет. Кадры в пакет могут заноситься до тех пор, пока не будет достигнута предельная величина пакета. Если она достигнута после того, как началась передача, узел может продолжить передачу до тех пор, пока не будет передан весь кадр.

Проблема использования витой пары 5 категории была решена несколько сложнее. Разработчики решили использовать все четыре витые пары кабеля, в этом случае скорость передачи по каждой паре уменьшатся до 250 Мбит/с.

Для передачи данных был применён код PAM5, использующий 5 уровней потенциала. Поэтому за один такт по одной паре передается 2,322 бита информации. Следовательно, тактовая частота с 250 МГц снижается до 125 МГц. Использование витой пары категории 5 на скорости 125 Мбит/с возможно, поскольку на заводах-изготовителях этот тип кабеля проходит тестирование на частоту 125 Мбит/с, именно этим обстоятельством и воспользовались разработчики стандарта. Если же использовать не все коды, а передавать 8 бит за такт (по 4 парам), то скорость передачи снижается до 108 Мбит/с и почти выдерживается требуемая скорость в 1000 Мбит/с. При этом остается запас неиспользуемых кодов, так как код PAM5 содержит 5^4=625 комбинаций, а для передачи за один такт по всем четырём парам 8 бит данных требуется всего 2^8=256 комбинаций. Оставшиеся комбинации приёмник может использовать для контроля принимаемой информации и выделения правильных комбинаций на фоне шума.

Для организации полнодуплексного режима разработчики применили технику, используемую при организации дуплексного режима на одной паре проводов в современных модемах. Информация обоими узлами одновременно передаётся по каждой из 4-х пар навстречу друг другу (рис. 5.12). При полудуплексном режиме работы получение встречного потока данных считается конфликтом, а для полнодуплексного режима работы – нормальной ситуацией.

Для выделения полезного сигнала из принимаемого сигнала используется метод вычитания – приёмник вычитает из результирующего сигнала

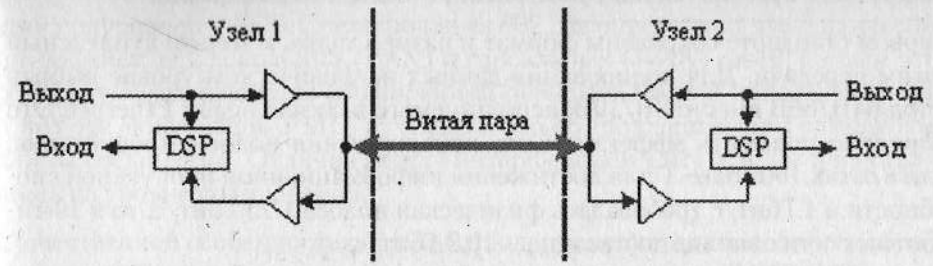


Рис. 5.12. Двунаправленная передача по витой паре

известный ему свой сигнал. Естественно, что это не простая операция и для её выполнения используются специальные цифровые сигнальные процессоры – DSP. На рис. 5.12 приведена схема организации передачи в полнодуплексном режиме по одной витой паре. Поскольку витых пар в кабеле 4, то на каждом узле необходимо установить 4 цифровых сигнальных процессора, со схемами гальванической развязки и усилителями. В первые годы появления стандарта Gigabit Ethernet такие сетевые карты стоили очень дорого, что собственно и препятствовало широкому распространению стандарта. В настоящее время, благодаря успехам микроэлектроники, все четыре DSP и сопутствующие схемы обработки информации выполняются в виде одной микросхемы, что способствует резкому снижению стоимости сетевых карт и, как следствие, широкому распространению стандарта Gigabit Ethernet.

Принятый стандарт Gigabit Ethernet на неэкранированной витой паре категории 5 позволяет использовать уже имеющуюся проводку и не требует её замены.

5.6. Высокоскоростная технология 10Gigabit Ethernet

Резкое увеличение количества передаваемой информации по сетям привело к необходимости разработки более скоростного стандарта. Деятельность по разработке стандарта 10Gigabit Ethernet началась в 1999 году с образования небольшой исследовательской группы, которая весной 2000 года была преобразована в рабочую группу 802.3ae. В феврале того же года её поддержали семь компаний – 3Com, Cisco, Extreme Networks, Intel, Nortel Networks, Sun и World Wide Packets, которые учредили консорциум 10Gigabit Ethernet Alliance (10GEA). Новая организация поставила перед собой двоякую цель: способствовать скорейшему принятию стандарта IEEE 802.3ae и активно пропагандировать новую технологию среди потенциальных потребителей.

В настоящее время в состав 10GEA входят около 110 компаний.

В новом стандарте сохранены формат и размер кадра, и только дуплексный режим передачи. Для кодирования данных на физическом уровне выбран метод 64B/66B вместо 8B/10B, используемого в сетях Gigabit Ethernet. Это позволило повысить эффективность использования полосы пропускания: если в сетях 1000Base-T для достижения информационной пропускной способности в 1 Гбит/с требовалась физическая полоса 1,25 Гбит/с, то в 10-гигабитных сетях вполне достаточно – 10,3 Гбит/с.

Платой за возросшую эффективность являются ограниченные возможности исправления ошибок передачи. Только внедрение алгоритма упреждающей коррекции ошибок и усовершенствованных методов восстановления сигналов позволило обеспечить надёжность сетей Ethernet на прежнем уровне ошибок – 10^{-12}.

Новый стандарт 10Gigabit Ethernet (первая версия которого была принята в 2002 году) включает в себя семь стандартов физической среды для LAN, MAN и WAN:

- **10GBase-CX4** – технология для коротких расстояний (до 15 метров), используется медный кабель CX4;
- **10GBase-SR** – технология для коротких расстояний (до 26 или 82 метров, в зависимости от типа кабеля), используется многомодовый оптический кабель. Он также поддерживает расстояния до 300 метров при использовании нового многомодового оптического кабеля;
- **10GBase-LX4** – технология, использующая уплотнение по длине волны для поддержки расстояний от 240 до 300 метров по многомодовому оптическому кабелю. Также поддерживает расстояния до 10 километров при использовании одномодового оптического кабеля;
- **10GBase-LR и 10GBase-ER** – эти технологии поддерживают расстояния до 10 и 40 км соответственно, при использовании одномодового оптического кабеля;
- **10GBase-SW, 10GBase-LW и 10GBase-EW** – эти технологии используют физический интерфейс, совместимый по скорости и формату данных с интерфейсом SONET/SDH. Они подобны стандартам 10GBase-SR, 10GBase-LR и 10GBase-ER, так как используют те же самые типы кабелей и расстояния передачи;
- **10GBase-T** – технология, использующая неэкранированную витую пару.

В архитектуре 10Gigabit Ethernet физический уровень разделяется на два

подуровня: зависящий от среды передачи подуровень – PMD и кодирующий подуровень – PCS. Спецификации 802.3ae определяют два физических интерфейса: для локальных (LAN) и глобальных (WAN) сетей, причём второй фактически является расширением первого. Оба они используют один и тот же подуровень PMD и, следовательно, поддерживают одинаковые длины соединений. Отличие между интерфейсами LAN и WAN существует лишь на уровне кодирования.

Существенной особенностью нового стандарта является способность оборудования 10Gigabit Ethernet взаимодействовать с сетями SONET/SDH, которая означает возможность передавать пакеты Ethernet по каналам SONET/SDH с высокой эффективностью.

Появление стандарта 10Gigabit Ethernet позволило технологию Ethernet распространить на глобальные сети, именно для этой цели предназначен интерфейс WAN. Поскольку полоса пропускания интерфейсов SONET/SDH мало отличается от полосы пропускания 10Gigabit Ethernet (9,29 и 10 Гбит/с соответственно), то разработка единого протокола MAC-уровня для двух физических интерфейсов с близкими значениями скорости передачи не вызвала особых затруднений.

В то же время в целях значительного удешевления будущего оборудования 10Gigabit Ethernet члены рабочей группы 802.3 отказались от достижения полной совместимости с сетями SONET/SDH. В частности, допустимая нестабильность задержки передачи, методы синхронизации и ряд оптических характеристик сетей SONET/SDH не получили отражения в стандарте 10Gigabit Ethernet. Массовому внедрению новой технологии пока препятствует высокая цена – не менее $3000 за порт.

5.7. Перспективы развития технологии Ethernet

Постоянный рост количества передаваемой информации в сетях приводит к необходимости поиска всё новых более скоростных технологий передачи информации, особенно по магистральным каналам. Этому способствует широкое распространение средств предоставления и обработки мультимедийной информации (например, цифровое кабельное телевидение, видеоконференции). Логично было бы предположить, что следующий стандарт будет иметь скорость передачи в 10 раз выше предыдущего – 100 Гбит/с. Наиболее подходящей средой для передачи информации со скоростями свыше 10 Гбит/с, обладающей необходимой помехозащищённостью и дальностью передачи, является оптическое волокно. Однако на пути использования

оптического волокна при таких скоростях необходимо решить много сложных задач.

Во-первых, передатчик информации – лазерный источник света. Чем чаще «моргает» (включается-выключается) лазер, тем больше данных можно передать по оптическому волокну. Однако технологические трудности производства лазеров ограничивают предельную скорость, с которой лазер может включаться и выключаться, генерируя настолько мощный квант света, чтобы его мощность на другом конце кабеля была достаточной для принимающей аппаратуры на уровне нескольких десятков гигагерц. По этой причине в настоящее время невозможно достичь скорости в 100 Гбит/с, используя один лазерный источник.

Один из способов увеличения количества передаваемой информации заключается в частотном уплотнении, особенность которого состоит в передаче нескольких длин волн в одном кабеле. Для реализации такого метода необходимо иметь несколько лазеров с разными частотами генерации света, образующих несколько информационных каналов, и систему оптического сведения лучей лазеров в один кабель. Например, компания Alcatel для увеличения пропускной способности кабеля использует 365 оптических каналов с шагом в 10 нм со скоростью 10 Гбит/с каждый, обеспечивая их передачу на расстояние 6800 км. Эта технология, предназначенная для подводных кабельных систем, способна обеспечивать пропускную способность, достаточную для передачи 47 млн. телефонных разговоров.

Как только сигналы достигают оконечного устройства, они должны быть разделены и отправлены получателям. Самый простой способ – направить одни и те же сигналы всем приёмникам. Однако каждый фотоприёмник должен воспринимать свет только определённой длины волны и игнорировать все остальные. Такие узкополосные приёмники стоят очень дорого. Поэтому чаще всего используют широкополосные приёмники, а селекцию длин волн осуществляют оптическими методами.

Один из способов выделения желаемой длины волны основан на использовании брэгговской решетки. Её можно представить в виде набора канавок, вытравленных поперек сердцевины оптоволокна. Если канавки расположены на определенном расстоянии друг от друга, то свет с некоторой длиной волны будет отражаться от такой решетки в обратном направлении. Затем решетка из косых "зарубок" отражает этот свет под углом, выводя его из волокна и направляя на приёмник. Используя эту технологию, можно разделить все длины волн и направить их на соответствующие приёмники.

Во-вторых, конечная задача любой оптической системы связи – это доставка световых импульсов из точки А в точку В, расположенных на требуемом

расстоянии. Интенсивность света в кабеле ослабевает, и для передачи информации на большие расстояния необходимо его усиливать (регенерировать).

Возможны два способа регенерации оптического сигнала. Первый работает по схеме оптика–электроника–оптика, а второй использует оптические усилители. Системы первого вида принимают оптический сигнал, преобразуют его в электрический, а затем электрические сигналы очищают от шума, усиливают и преобразуют обратно в оптический.

Оптические усилители усиливают сигнал без преобразования его в электрический. Наибольшее распространение получили усилители на волокне, легированном редкоземельным элементом: эрбием, празеодимом или иттербием. В таком усилителе на участке, легированном редкоземельным элементом, полезный сигнал смешивается с излучением лазера накачки, работающего на длине волны 980 или 1480 нм. Лазер накачки возбуждает атомы редкоземельного элемента, и, когда на них падает свет полезного сигнала, часть энергии возбуждения передается этому сигналу, усиливая его.

Такие усилители позволяют значительно снизить потребность в дополнительном оборудовании и не добавляют заметной задержки в проходящий сигнал. Однако вместе с полезным сигналом усиливают и шум в оптическом сигнале, что на больших расстояниях может привести к серьезным искажениям.

В-третьих, для оптических сетей с такой скоростью передачи целесообразно использовать чисто оптические системы коммутации – коммутаторы, что позволит не только увеличить пропускную способность коммутатора, но и значительно снизить уровень оптического шума. Компания Opthos уже разработала опытные образцы оптических коммутаторов, к сожалению, они пока крайне дороги.

5.8. Технология 40Gigabit Ethernet

При разработке стандарта 40Gigabit Ethernet разработчики стремились:

- сохранить формат кадров Ethernet;
- сохранить минимальные и максимальные размеры кадров;
- обеспечить уровень ошибок не более 10^{-12}.

Новый стандарт 40Gigabit Ethernet (окончательная версия которого была принята в 2010 году) включает в себя четыре стандарта физической среды:

- **40GBase-KR4** – технология для коротких расстояний (до 1 метра), по проводникам печатной платы устройств;
- **40GBase-CR4** – технология для коротких расстояний (до 10 метров), использующая медный кабель CX4;
- **40GBase-SR4** – технология для передачи информации на расстояние не менее 100 метров по многомодовому оптическому кабелю;
- **40GBase-LR4**– технология для передачи информации на расстояние не менее 10 км по одномодовому оптическому кабелю.

Однако разработчики не стали себя особо утруждать разработкой стандарта для сверхбыстродействующих устройств, а пошли по пути наименьшего сопротивления. Так, передача информации на 1 и 10 метров по медному проводу осуществляется при помощи объединения 4-х стандартных линий 10GBase-CX4. А для передачи на 10 км по одномодовому кабелю используется метод частотного уплотнения – 4 волны разной длины (около 1310 нм). В результате, хотя скорость передачи информации и возросла до 40 Гбит/с, с точки зрения развития технологии данный стандарт не предлагает ничего нового.

5.9. Технология 100Gigabit Ethernet

Как было отмечено в п. 7 данной главы, разработка стандарта Ethernet для передачи информации со скоростью свыше 10 ГГбит/с сопряжена с колоссальными технологическими проблемами, которые в полной мере до сих пор преодолеть не удалось. Максимальная скорость, которую удалось достичь для одной линии, не превышает 25 – 30 ГГбит/с. Поэтому разработчики стандарта 100Gigabit Ethernet пошли по уже проторённому пути, который они применили при разработке стандарта 40Gigabit Ethernet. В результате новый стандарт (окончательная версия которого была принята в 2010 году) оговаривает четыре типа физической среды:

- 100GBase- CR4 – технология для коротких расстояний (до 10 метров), использующая медный кабель CX4;
- 100GBase- SR4 – технология для передачи информации на расстояние не менее 100 метров по многомодовому оптическому кабелю;
- 100GBase- LR4 – технология для передачи информации на расстояние не менее 10 км по одномодовому оптическому кабелю;
- 100GBase-LR4– технология для передачи информации на расстояние не менее 40 км по одномодовому оптическому кабелю.

Данный стандарт использует те же принципы объединения 4-х линий или частотное уплотнение, что и стандарт 40Gigabit Ethernet, только скорость передачи по отдельной линии увеличена до 25 ГГбит/с.

В результате, хотя скорость передачи информации и возросла до 100 Гбит/с, но с точки зрения развития технологии данный стандарт оговаривает увеличение скорости передачи информации всего в 2,5 раза, по сравнению со стандартом 10Gigabit Ethernet. Таким образом, за 8 лет, прошедших с момента появления стандарта 10Gigabit Ethernet, технологию передачи данных удалось улучшить всего в 2,5 раза.

Тем не менее, появление стандарта 100Gigabit Ethernet привело к тому, что начиная с 2011 года все ведущие поставщики сетевого оборудования приступили к серийному выпуску 100 Гбитных маршрутизаторов.

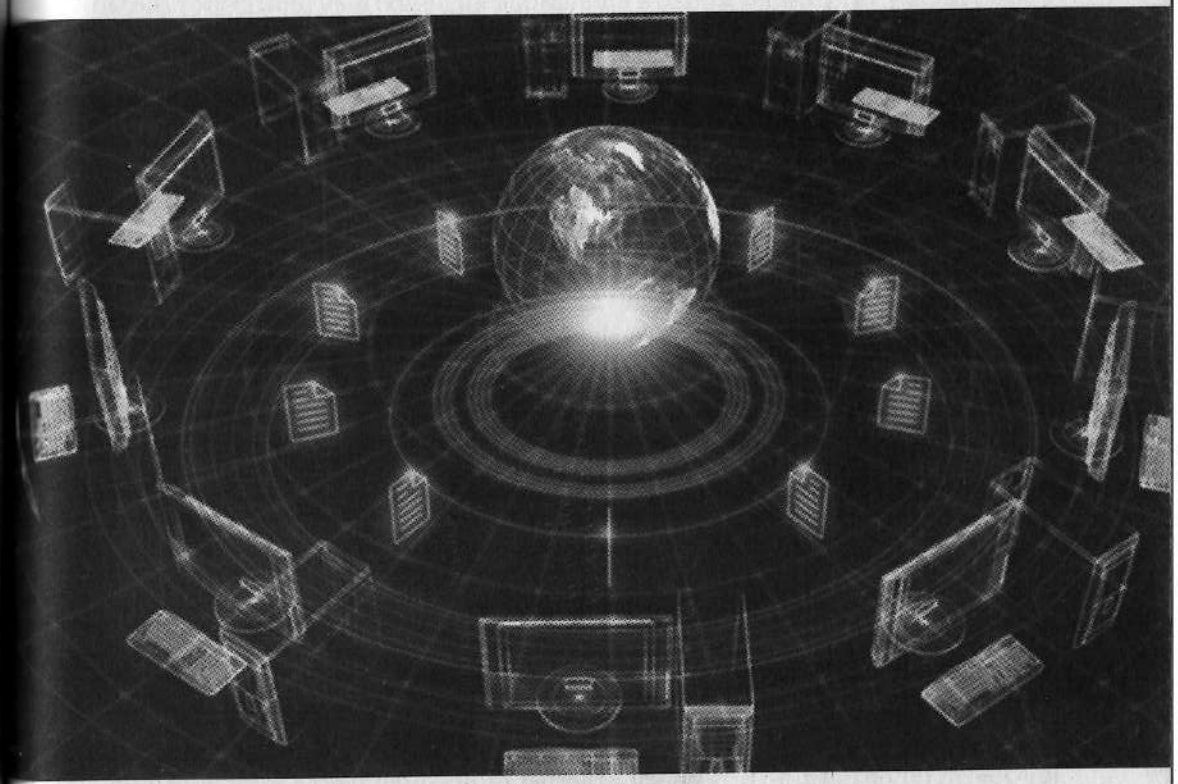
Контрольные вопросы

1. *Что такое MAC-адрес?*
2. *Какой метод доступа к передающей среде используется в технологии Ethernet?*
3. *Дайте определение конфликтной ситуации в Ethernet.*
4. *Каким образом происходит определение конфликтной ситуации?*
5. *Что такое пауза и как она вычисляется в методе доступа CSMA/CD?*
6. *Каким образом разрешаются конфликты в технологии Ethernet?*
7. *Поясните понятие "время двойного оборота".*
8. *Какова производительность сети Ethernet при передаче пакетов минимальной длины?*
9. *Какова производительность сети Ethernet при передаче пакетов максимальной длины?*
10. *Назовите основные характеристики "толстого Ethernet".*
11. *Назовите основные характеристики "тонкого Ethernet".*
12. *Назовите основные характеристики Ethernet на витой паре.*
13. *Объясните отличия полудуплексного режима от полнодуплексного.*
14. *Назовите основные характеристики технологии Fast Ethernet.*
15. *Назовите достоинства и недостатки коаксиального кабеля.*
16. *Назовите достоинства и недостатки витой пары.*
17. *Назовите достоинства и недостатки оптического кабеля.*

18. Перечислите основные характеристики кабеля – витой пары известных Вам категорий.
19. Объясните, почему произошёл переход на витую пару.
20. Объясните, каким образом удалось увеличить диаметр сети Gigabit Ethernet до 200 метров.
21. Объясните, каким образом удалось предать 1000 Мбит в секунду по витой паре 5 категории.
22. Какой тип передающей среды используется в 10 Gigabit Ethernet?
23. Почему не удалось создать 100 Gigabit Ethernet?
24. Какова в настоящее время максимальная скорость передачи информации в технологии Ethernet?
25. Назовите отличия Fast Ethernet от Ethernet.
26. Какие методы кодирования используются в Gigabit Ethernet?
27. Какой метод кодирования принят в 10 Gigabit Ethernet?
28. В каких случаях целесообразно применять технологию 10 Gigabit Ethernet?
29. Какие проблемы необходимо решить для реализации Ethernet со скоростью более 10 Gigabit?
30. За счет чего можно увеличить диаметр сети?

Глава 6.

КОММУТАТОРЫ



Несомненное достоинство разделяемой среды передачи данных заключается в простоте топологии среды, допускающей лёгкое наращивание числа узлов (в небольших пределах), а также простоте протоколов, обеспечивающих низкую стоимость сетевых адаптеров, повторителей и концентраторов.

Несмотря на перечисленные достоинства, разделяемая среда имеет очень существенный недостаток. Увеличение количества узлов, или интенсивности передач узлами, или и то и другое приводит к резкому снижению пропускной способности сети при превышении некоторого порога количества узлов, подключенных к разделяемой среде. Причина заключается в случайном характере метода доступа к среде, используемом во всех технологиях Ethernet. Технологии Ethernet присущ экспоненциальный рост величины задержек доступа (рис. 6.1). Порог, при котором наступает резкий перелом в поведении сети, находится на уровне 30 – 50 узлов.

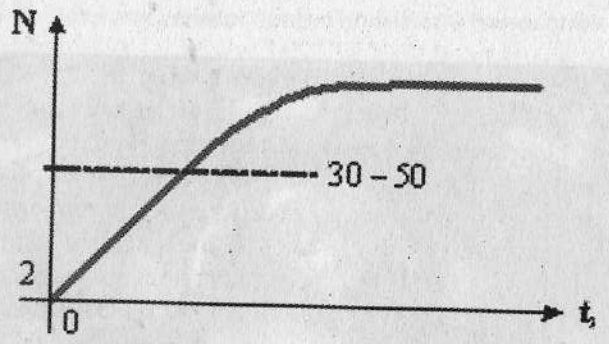


Рис. 6.1. Зависимость времени задержки кадра от числа узлов

При загрузке сети, таким образом, когда время задержки в передаче кадра увеличивается до 1,5 раз, технология Ethernet хорошо справляется с передачей информации. Однако при повышении интенсивности передач сеть всё больше времени начинает проводить неэффективно, повторно передавая кадры, которые вызвали конфликт. При возрастании интенсивности передач практически любой кадр сталкивается с другими кадрами, вызывая конфликт, при этом время задержки tз увеличивается настолько, что вызывает недовольство всех пользователей, и принципиально стремится к бесконечности. Сеть перестает передавать полезную информацию и работает «на себя», обрабатывая конфликты.

Этот эффект хорошо известен на практике и исследован путём имитационного моделирования, поэтому сети Ethernet не рекомендуется загружать так,

чтобы среднее время передачи пакета возрастало более чем в 3 раза. Более удобной величиной оценки пропускной способности сети является коэффициент её использования, который представляет собой отношение количества передаваемой информации в единицу времени к максимальной пропускной способности сети. Именно поэтому во многих системах управления сетями пороговая граница для индикатора коэффициента загрузки сети Ethernet по умолчанию устанавливается на величину 30 %.

При достижении сетью заданного коэффициента использования администраторы должны принять все доступные им методы к уменьшению задержек, иначе все достоинства сети будут сведены на нет крайне низкой производительностью.

6.1. Мосты

С целью уменьшения количества конфликтов и повышения эффективности работы сети целесообразно разделить её на части – сегменты, так чтобы внутрисегментный обмен информацией в одном сегменте не мешал обмену в другом. Разделение сети можно произвести простым физическим разделением сети на части. Правда, в этом случае для передачи информации из одного сегмента в другой необходимо использовать внешние носители информации – дискеты, флеш-память и т.д. Однако такой способ обмена несколько неудобен, поэтому для деления сети на части разработчики оборудования придумали устройство под название мост. Мосты незаметны для узлов, поскольку они самостоятельно строят специальную адресную таблицу, на основании которой решают, нужно ли передавать пришедший кадр в другой сегмент.

Мост строит свою адресную таблицу на основании пассивного наблюдения за кадрами, циркулирующими в подключённых к его портам сегментах. При этом мост учитывает адреса источников кадров данных, поступающих на порты моста. По адресу источника кадра мост делает вывод о принадлежности этого узла тому или иному сегменту сети. Рассмотрим процесс автоматического создания адресной таблицы моста и её использования на примере простой сети, представленной на рис. 6.1.

Мост соединяет два логических сегмента. Сегмент 1 составляют узлы, подключённые с помощью одного отрезка коаксиального кабеля к порту 1 моста, а сегмент 2 – узлы, подключённые с помощью другого отрезка коаксиального кабеля к порту 2 моста.

Каждый порт моста работает как конечный узел своего сегмента, за одним исключением – порт моста не имеет собственного адреса. Порт моста

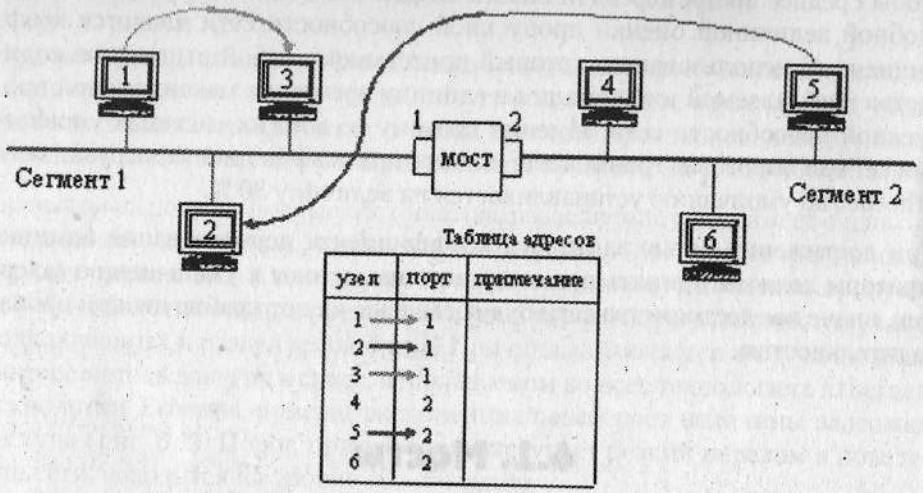


Рис. 6.2. Принцип работы моста

следит за всеми пакетами, передаваемыми в присоединённых к нему сегментах, и использует проходящие через него пакеты для изучения состава сети. Поскольку в буфер записываются все пакеты, то адрес порта мосту не нужен.

В исходном состоянии мост ничего не знает о том, какие узлы подключены к каждому из его портов. Поэтому мост просто передаёт кадры с одного порта на другой и наоборот. Кадр мост передаёт не побитно, а с буферизацией, что разрывает логику работы всех сегментов как единой разделяемой среды, поэтому, когда мост собирается передать кадр с сегмента на сегмент, он заново пытается получить доступ к сегменту как узел сети по правилам CSMA/CD.

Одновременно с передачей кадра мост определяет адрес источника кадра и делает новую запись о его принадлежности в своей адресной таблице. Например, получив на свой порт кадр от узла 2, мост делает первую запись в своей адресной таблице: адрес 2 – порт 1. Если все шесть узлов данной сети проявляют активность и посылают друг другу кадры, то скоро мост построит полную адресную таблицу сети, состоящую из 6 записей – по одной записи на узел.

После того как мост прошёл этап обучения, он может работать более рационально. При получении кадра, направленного от узла 1 к узлу 3, он просматривает адресную таблицу на предмет совпадения адресов с адресом назначения 3. Поскольку такая запись есть, то мост выполняет второй этап анализа таблицы – проверяет, находятся ли узлы с адресами источника (в нашем случае – это адрес 1) и адресом назначения (адрес 3) в одном сегменте. Поскольку они находятся в одном сегменте, то мост не передаёт кадр на другой порт, а просто удаляет его из буфера.

При получении кадра от узла 5, адресованного узлу 2, мост на основании своей таблицы адресов определяет, что они находятся в разных сегментах. Поэтому он начинает процесс передачи кадра в сегмент 1, конкурируя с узлами этого сегмента за доступ к среде.

Таблица адресов может формироваться динамически, в процессе самообучения моста, или вручную администратором сети. Динамические записи имеют срок жизни – при создании или обновлении записи с ней связывается отметка времени, которая помещается в поле – примечание. Это даёт возможность автоматически реагировать на перемещения узла из сегмента в сегмент. После включения этого узла в работу в другом сегменте его кадры начнут попадать в буфер моста через другой порт, и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Статические записи не имеют срока жизни, что позволяет администратору «подправлять» работу моста, если это необходимо. Кадры с широковещательными адресами передаются мостом на все порты.

Если после применения моста производительность сети не достаточна, то для повышения производительности сети можно установить ещё один или несколько мостов, разделив сеть на несколько сегментов. Рассмотрим в качестве примера сеть, представленную на рис. 6.3. В данном примере, для повышения производительности, администратор установил два моста па-

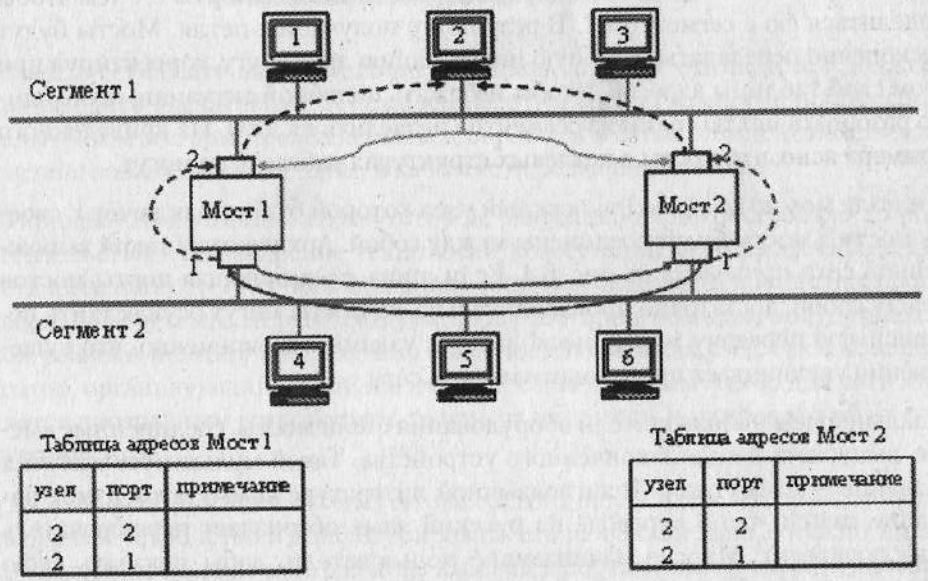


Рис. 6.3. Ограничения сети, построенной на мостах

раллельно, в надежде на то, что, по аналогии с рекой, два моста через реку обеспечивают увеличение потока машин минимум в два раза.

Допустим, в сегменте 1 сети появился новый узел под номером 2. Он решил заявить о себе, рассылкой широковещательного сообщения всем узлам сети. Это сообщение приняли оба моста. Естественно, что каждый мост в своей таблице адресов сделал соответствующую запись (2 – 2) с пометкой о времени её создания в поле примечание. Поскольку полученное сообщение широковещательное, то каждый мост должен передать его в другой сегмент (сегмент 2), через свой первый порт всем остальным узлам сети. Поскольку сегменты представляют собой сеть Ethernet с разделяемой средой, то мосты начинают конкурировать между собой за обладание общим каналом связи.

Предположим, что мосту 1 удалось первому захватить канал связи. Он пересылает полученное от узла 2 сообщение. Все узлы сегмента 2 получают это сообщение. Его также получает и мост 2. Получив это сообщение, мост 1 понимает, что узел 2 подключён к порту 1, однако запись в его таблице адресов говорит о том, что узел 2 подключен к порту 2. Поскольку полученное сообщение более свежее, то мост 1 удаляет старую запись "2 – 2" и на её место помещает более "свежую", а стало быть, правильную запись "2 – 1".

Аналогично поступает и мост 2, когда мост 1 пересылает ему копию сообщения от узла 2, он корректирует запись в таблице адресов.

Затем они оба должны новую информацию передать на порты 2, с тем чтобы поделиться ею с сегментом 1. В результате получается петля. Мосты будут бесконечно передавать подобную информацию друг другу, корректируя при этом свои таблицы адресов. Чтобы избежать подобной ситуации, необходимо разорвать петлю (один из сегментов разделить на два). Из приведённого примера ясно, что мосты в петлевых структурах работать не могут.

В идеале можно создать сеть, каждый узел которой будет подключён к своему мосту, а мосты будут соединены между собой. Архитектура такой вырожденной сети приведена на рис. 6.4. Если шина, соединяющая порты мостов между собой, достаточно производительна, то мосты могут осуществить независимую передачу между несколькими узлами одновременно, что существенно увеличивает производительность сети.

В дальнейшем производители оборудования стали мосты, соединённые вместе, выпускать в виде законченного устройства. Такой мультимост получил название – коммутатор. В англоязычной литературе коммутатор имеет название switch, что в переводе на русский язык обозначает переключатель (выключатель). Многие начинающие пользователи, дабы показать свою грамотность (учёность) на фоне остальных, в своём лексиконе используют иностранное слово - свитч, хотя сами толком не знают, зачем он нужен и что

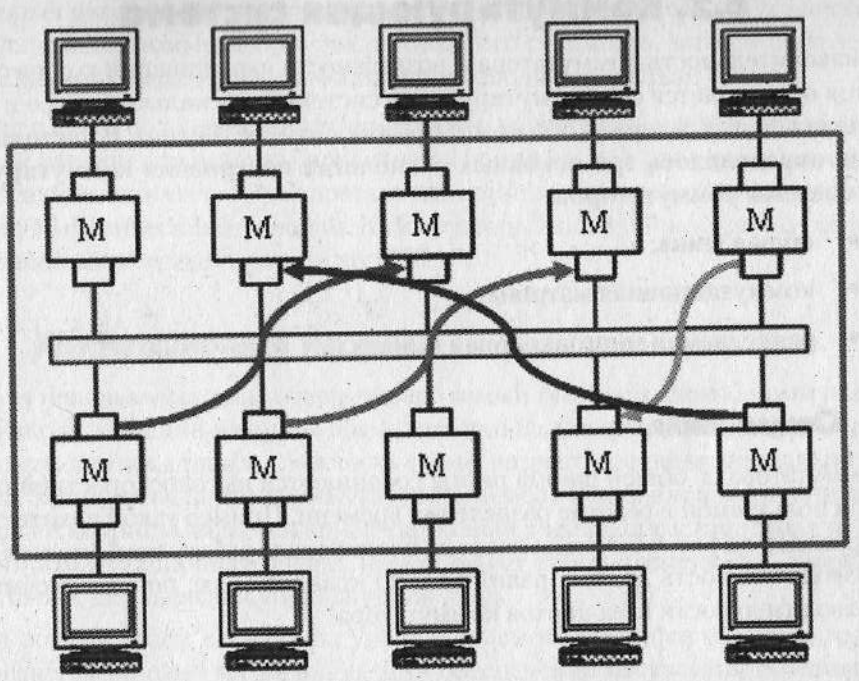


Рис. 6.4. Архитектура вырожденной сети

делает в сети. Зато на своих соплеменников, как они считают, это должно производить впечатление. К сожалению, этим грешат и многие профессионалы. Коммутаторы предназначены для работы в сетях с топологией звезда с использованием витой пары, в качестве передающей среды.

Широкому применению коммутаторов, безусловно, способствовало то обстоятельство, что внедрение технологии коммутации не требовало замены установленного в сетях оборудования – сетевых адаптеров, концентраторов, кабельной системы. Порты коммутаторов работали в обычном полудуплексном режиме, поэтому к ним можно было подключить как узел, так и концентратор, организующий целый логический сегмент. Коммутатор для сети является прозрачным устройством, то есть узлы даже не подозревают об их существовании.

Удобство использования коммутатора состоит ещё и в том, что это самообучающееся устройство и конфигурировать его не обязательно – нужно только правильно подключить разъёмы кабелей к портам коммутатора, а дальше он будет работать самостоятельно и эффективно выполнять поставленную перед ним задачу повышения производительности сети.

6.2. Коммутирующая система

Производительность коммутатора и возможность наращивания количества портов определяется его коммутирующей системой, насколько быстро и без дополнительных задержек она может производить соединение. **В настоящее время определилось три основных технологии построения коммутирующей системы коммутаторов:**

- **общая шина;**
- **коммутационная матрица;**
- **разделяемая многовходовая память.**

Общая шина

В коммутаторах с общей шиной порты соединяются высокоскоростной шиной, используемой в режиме разделения времени. Пример такой архитектуры приведён на рис. 6.5. Чтобы шина не блокировала работу коммутатора, её производительность должна равняться, по крайней мере, половине суммы производительности всех портов коммутатора.

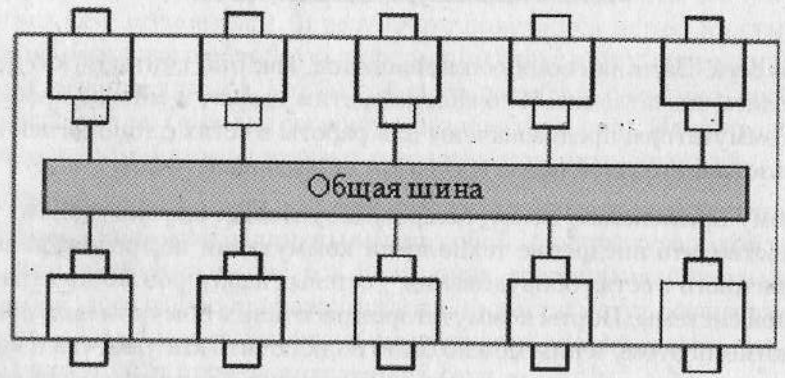


Рис. 6.5. Коммутирующая система – общая шина

В коммутаторах, использующих в качестве коммутирующей системы общую шину, для уменьшения непроизводительных простоев шины отказались от случайного метода доступа. В таких коммутаторах применяют арбитр шины,

который регулирует очерёдность доступа к передающей среде. Это несколько усложняет коммутатор и увеличивает его стоимость, зато лишние затраты с лихвой окупаются высочайшей производительностью.

Общая шина позволяет достаточно просто наращивать производительность путём увеличения ширины последней. Например, для увеличения производительности шины в 16 раз достаточно организовать передачу одновременно по 16 физическим проводам. Коммутаторы с общей шиной выпускают на большое количество портов – до 1024.

Коммутационная матрица

Коммутационная матрица обеспечивает самый быстрый способ коммутации. Однако реализация матрицы возможна только для относительно небольшого числа портов, причём сложность схемы возрастает пропорционально квадрату количества портов коммутатора. Элементарным элементом коммутирующей матрицы является коммутационная ячейка, представленная на рис. 6.6 *а*. Коммутационная ячейка представляет собой элемент 2x2, имеющий два входа, два выхода и управляющий вход.

Под воздействием сигнала на управляющем входе информация со входов передаётся на один из выходов. Для обеспечения отсутствия смешивания

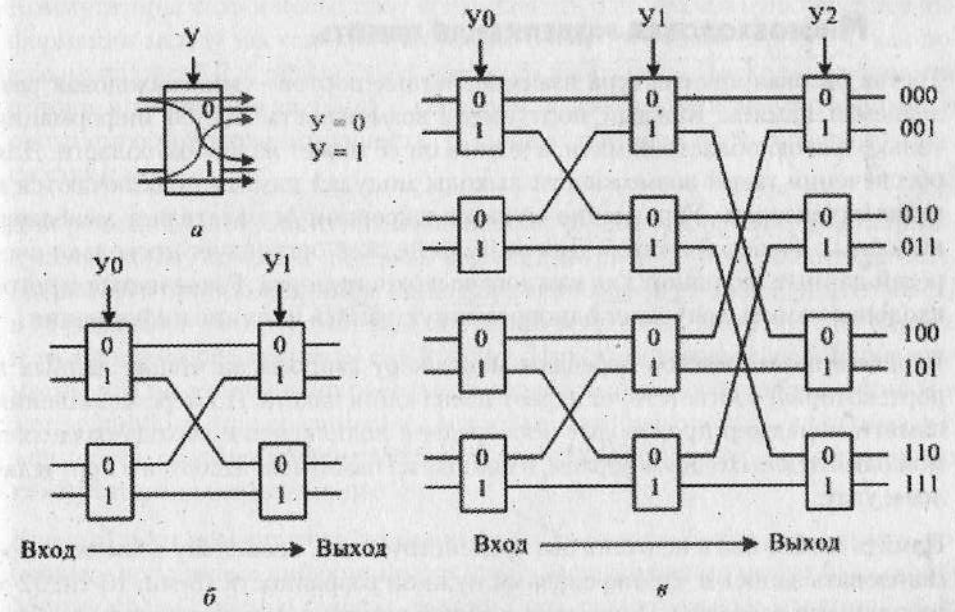


Рис. 6.6. Коммутирующие матрицы: а) элементарная ячейка; б) 4 × 4; в) 8 × 8

сигналов на выходе информация должна присутствовать только на одном из входов.

Из подобных ячеек можно построить коммутационную матрицу большего размера, например 4 x 4 или 8 x 8, как показано на рис. 6.6. Входные порты на основании просмотра адресной таблицы коммутатора определяют по адресу назначения номер выходного порта. Для данного примера поле адреса представляет собой просто 2-х или 3-х разрядное двоичное число, соответствующее номеру выходного порта. Эта информация передаётся в очередь арбитра коммутационной матрицы, который в соответствии с ней организует управление матрицей и осуществляет соединение портов.

В приведённом примере матрица состоит из двух или трёх уровней элементарных коммутирующих ячеек, которые соединяют свой вход с одним из двух выходов в зависимости от значения поля адреса.

Достоинством коммутационной матрицы является её высочайшее быстродействие и регулярная структура, которую удобно реализовывать в интегральных микросхемах. Зато после реализации матрицы N x N в составе БИС возникает проблема с дальнейшим наращиваем портов, поэтому коммутаторы на коммутационной матрице имеют ограниченное количество портов, обычно не более 64.

Многовходовая разделяемая память

Третья базовая архитектура взаимодействия портов – многовходовая разделяемая память. Каждый порт имеет возможность записи информации только в свою область памяти, а читать он её может из любой области. Для обеспечения такой возможности выходы модулей памяти подключаются к мультиплексорам. Управление мультиплексорами осуществляет менеджер выходных очередей портов. Для этого менеджер организует несколько очередей данных, по одной для каждого выходного порта. Разделяемая многовходовая память допускает одновременную запись и чтение информации.

Входные порты мостов передают менеджеру запросы на чтение данных в порт, который соответствует адресу назначения пакета. По мере заполнения памяти менеджер производит поочередное подключение выхода разделяемой памяти к выходным портам, и данные из памяти передаются в порт и далее к узлу.

Память может быть не очень быстродействующей, поскольку возможно организовать запись и чтение словами нужной разрядности (8-ми, 16-ти, 32-х разрядными словами). Применение общей памяти позволяет портам вообще отказаться от индивидуальной буферной памяти или, в некоторых случаях,

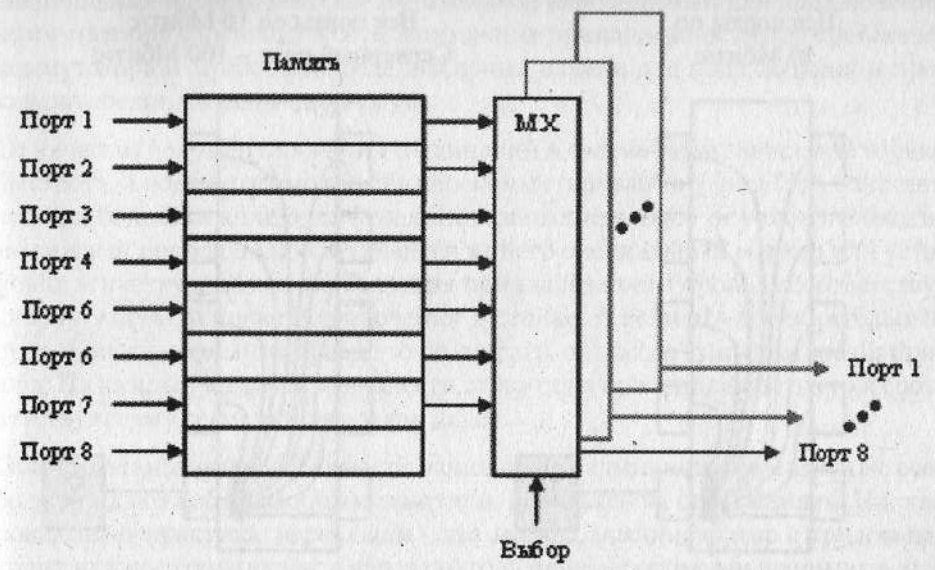


Рис. 6.7. Коммутация на основе многовходовой разделяемой памяти

резко снизить требования к размеру буферной памяти процессора порта.

Коммутаторы хоть и позволяют организовать одновременную передачу информации между несколькими портами, однако в сетях с сервером, как показано на рис. 6.7 а, они не дают никакого выигрыша. Поскольку основные потоки информации связаны с сервером, а все порты коммутатора имеют одинаковую производительность. В данном примере порт сервера является узким местом.

Для решения такой, часто возникающей, проблемы разработчики стали выпускать коммутаторы с портами, имеющими разную производительность. На рис 6.8 *б* приведён пример, когда порт сервера имеет производительность в 10 раз выше, чем все остальные порты коммутатора. Это позволяет за одно и то же время обслуживать до 10 узлов. При этом каждый узел будет «думать», что он монопольно работает с сервером. Аналогичная ситуация может возникнуть при соединении двух коммутаторов, организующих два сегмента сети, поэтому количество быстродействующих портов в современных коммутаторах всегда не менее 2-х.

Каждый порт коммутатора должен иметь некоторую буферную память, поскольку при приёме информации от узла узел назначения может быть занят. Минимальный размер буфера должен быть равен размеру максимального пакета. Однако обычно сообщение передаётся несколькими пакетами, поэтому порт коммутатора должен иметь достаточно вместительный буфер. В

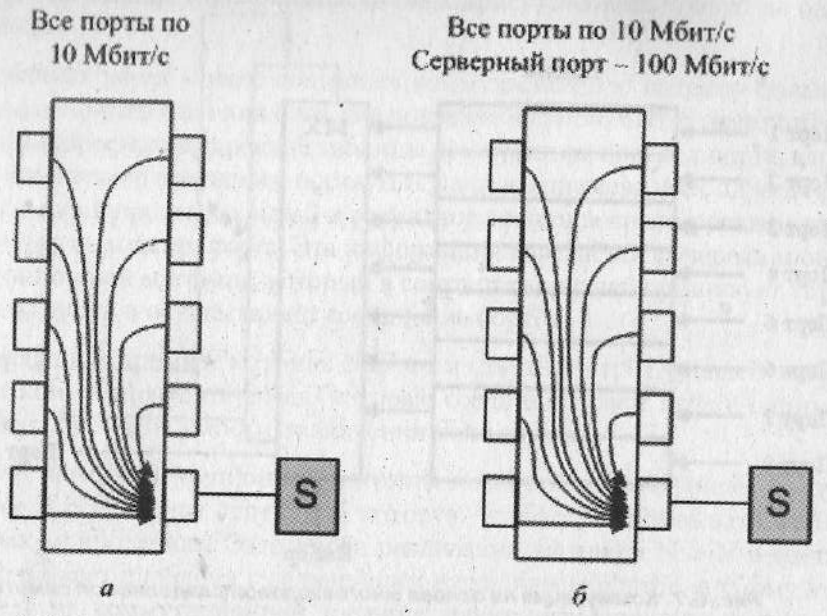


Рис. 6.8. Производительность портов коммутатора: а) одинаковая; б) разная

современных коммутаторах объём буфера порта может достигать нескольких сотен Кбайт, а в некоторых случаях и нескольких Мбайт.

Каждый порт имеет индивидуальную таблицу адресов. Вследствие того, что к порту может быть подключён другой коммутатор или повторитель, то порт в таблице адресов должен иметь возможность записи информации обо всех подключенных к данному порту узлах. Максимальное количество узлов в сети Ethernet равняется 1024, поэтому размер таблицы адресов порта должен позволять запомнить, как минимум, все адреса сети. В случае использования коммутатора в качестве магистрального – соединяющего несколько сетей, объём таблицы адресов должен позволять запоминать адреса узлов всех подключённых к нему сетей. Поэтому объём таблицы адресов может достигать 100 000 записей и более.

6.3. Конструктивное выполнение коммутаторов

Поскольку все коммутаторы являются системными устройствами, организующими передачу информации между узлами, то их желательно устанавливать подальше от любопытных пользователей. Для этих целей выпускаются

специальные шкафы, которые имеют набор необходимый для подключения коммутаторов к питающей сети, монтажные принадлежности для крепления коммутатора в стойку, распределительные панели для подключения и прокладки соединительных проводов.

От качества документирования соединений в дальнейшем зависит не только быстрота, а вообще возможность перекоммутирования узлов. Это означает, что при большом количестве соединительных проводов от узлов необходимо каждый провод подписать, нанеся на него специальный маркер или установив этикетку с целью определения принадлежности провода соответствующему узлу или точке подключения в стойке. А если это соединительный провод между коммутаторами, то подписать, от какого коммутатора он пришёл. На каждый коммутатор нанести этикетку о принадлежности его к соответствующему сегменту сети и так далее.

Обычно ленивые и в особенности начинающие администраторы сети не особо утруждают себя подобным занятием, полагаясь на свою память. Но, как показывает практика, через один - два месяца администратор с трудом находит нужное соединение, а через полгода он не в состоянии вспомнить, что он тут «нагородил». В случае же ухода такого горе-администратора с работы разобраться в его «зверинце» крайне сложно, это потребует больших усилий и времени.

Отдельно следует обратить внимание на культуру прокладки соединительных проводов в стойках. Многие администраторы не утруждают себя аккуратной прокладкой проводов в соответствующие держатели, а «временно» подключают узел к порту коммутатора. Но ничего нет постоянней временного. Поэтому со временем стойка обрастает неимоверным количеством свисающих проводов, которые полностью закрывают доступ к коммутаторам. Найти нужное соединение в такой «бороде» крайне сложно, нудно и утомительно. Если стойка снабжена ещё и дверью, то при закрывании двери обязательно часть проводов попадает под дверь, что может привести к натягиванию проводов и ухудшению контакта, а может быть даже перекушена ею. Найти причину падения производительности или перекушенный провод под оболочкой очень не простое дело.

Все производимые коммутаторы выпускаются в трёх основных конструктивных решениях: законченное устройство, стековый коммутатор, модульный коммутатор. Рассмотрим каждое из этих конструктивных решений более подробно.

Законченное устройство

Законченное устройство представляет собой коммутатор на определённое

количество портов, выполненный в виде прямоугольного ящика с соответствующим количеством разъёмов. Такой коммутатор имеет собственный источник питания и все необходимые схемы управления. На рис. 6.9 показана передняя панель такого коммутатора.

Рис. 6.9. Коммутатор Cisco серии 2960

Коммутатор имеет 24 порта со светодиодными индикаторами работы портов. В левой части коммутатора располагается несколько индикаторов, показывающих состояние коммутатора в данный момент времени. Каждый порт (разъём) имеет два индикатора. Один индикатор показывает, что к данному порту подключён узел и он активен. Второй индикатор загорается в моменты приёма или передачи пакета портом. Поскольку пакет передаётся достаточно быстро (от 67,1 мкс до 1 мс), то, если индикатор будет гореть только во время передачи пакета, ни один человек этого не увидит, поэтому он горит намного дольше и его «моргание» совершенно не соответствует количеству переданных пакетов. Поэтому «моргание» индикатора просто говорит о том, что через порт проходят пакеты.

Вообще наличие любой индикации благотворно влияет на обслуживающий персонал, поскольку «чёрный ящик» без индикации человеком воспринимается не совсем дружелюбно.

В правой части передней панели коммутатора находятся два разъёма для соединения коммутаторов между собой. Необходимость в наличии специальных разъёмов для соединения коммутаторов обусловлена тем, что порт коммутатора использует одни провода для передачи информации, а другие для приёма, а узел эти провода использует наоборот (рис. 6.10).

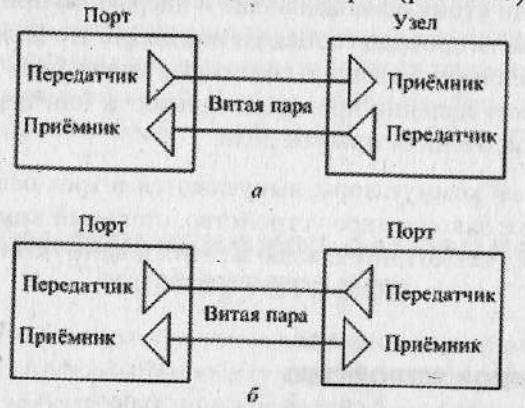


Рис. 6.10. Подключение: а) узел-порт; б) порт – порт

При подключении порта к порту тем же кабелем, что порт – узел, два передатчика будут работать навстречу друг другу, что приведёт к немедленному выходу из строя обоих, а приёмники при этом ничего не примут. Для того чтобы избежать такой ситуации, необходимо перемять «перевёрнутый» кабель. Это, конечно, хорошо, но если им случайно подключить порт – узел, то передатчики этих двух устройств тоже выйдут из строя. Чтобы избавиться от «перевёрнутого» кабеля и иметь возможность соединения двух коммутаторов в коммутаторе, и предусмотрены два разъёма – один как у узла, а другой «перевёрнутый», такое решение позволяет соединить коммутатор ещё с двумя коммутаторами. Коммутаторы, выпускаемые в виде законченного устройства, имеют не более 48 портов при стоимости подключения в районе $50 – $100 за порт.

Стековый коммутатор

Стандарт Ethernet на витой паре оговаривает максимальное количество повторителей (коммутаторов) между любыми двумя узлами на уровне 4-х. По мере роста сети часто возникает необходимость в подключении новых узлов. Если все порты коммутаторов заняты, то необходимо устанавливать ещё один коммутатор, а это может привести к выходу за границы стандарта, и коммутатор окажется 5-м.

Для решения подобных вышеописанных проблем промышленность выпускает коммутаторы, допускающие наращивание, при этом несколько соединённых вместе коммутаторов рассматриваются как один коммутатор. Такие коммутаторы получили название – стековые. Для объединения нескольких коммутаторов в стек используется специальный разъём, размещённый на задней панели коммутатора (рис. 6.11). С помощью этого разъёма внутренняя общая шина одного коммутатора соединяется с внутренней шиной другого коммутатора соответствующим кабелем. В результате получается один коммутатор, конструктивно выполненный в нескольких физических устройствах.

Современные коммутаторы допускают объединение до 9 устройств в стек. Но в связи с тем, что соединительный кабель стоит очень дорого, то стоимость одного подключения обходится несколько дороже и составляет около $60 - $120 за порт.

При включении питания коммутаторы определяют, что к разъёмам расширения подключен кабель, и автоматически настраиваются на работу в стеке. Дополнительной настройки коммутаторов для работы в стеке не требуется. Использование стековых коммутаторов оправдано в тех случаях, когда в будущем предполагается расширение сети, а средств на начало строительства сети недостаточно.

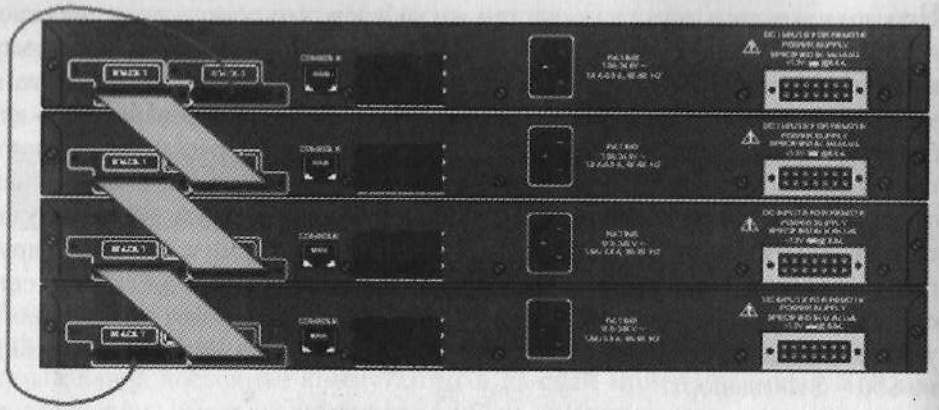


Рис. 6.11. Объединение коммутаторов в стек

Модульный коммутатор

Модульный коммутатор представляет собой шасси, в которое могут быть установлены модули в нужном количестве и требуемой производительности (рис. 6.12). Поскольку шасси содержит одно общее устройство управления и источник питания, то такой коммутатор имеет самую низкую стоимость подключения – около $30 за порт. Но нужно иметь в виду то обстоятельство, что приведенная цена достигается при установке всех модулей.

Модульный коммутатор имеет смысл использовать в том случае, если требуется большое количество портов, или большая номенклатура модулей, или и то и другое. Поскольку модульный коммутатор организует работу практически всей сети (они выпускаются на большое количество портов – до 1024), то его отказ вызовет остановку всей сети. Поэтому к такому коммутатору предъявляются повышенные требования по надёжности, в частности устанавливаются два источника питания, допускающих горячую замену. Также могут быть установлены два устройства управления, при отказе одного из них второе автоматически принимает на себя функции управления. Кроме этого, модульные коммутаторы имеют специальные направляющие для укладки сотен проводов, как показано на рис. 6.12.

Кроме конструктивного исполнения, выпускаемые коммутаторы имеют разные скорости передачи данных портов:

- 10 Мбит/с,
- 10 – 100 Мбит/с,
- 100 Мбит/с,
- 100 – 1000 Мбит/с,

- 1 Гбит/с,
- 1 – 10 Гбит/с,
- 10 Гбит/с,
- 10 – 40 Гбит/с,
- 40 Гбит/с.

Конструктивное исполнение, допускающее установку различного количества портов, имеющих разнообразные скорости и среды передачи, и наличие дополнительных функций позволяет иметь огромную номенклатуру выпускаемых коммутаторов, существенно отличающихся по цене.

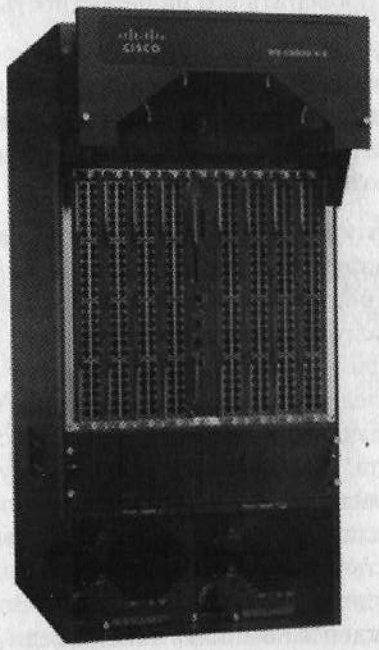


Рис. 6.12. Модульный коммутатор на 384 порта

Здесь следует заметить, что некоторые производители стали выпускать устройства под названием HUB – Switch (повторитель – коммутатор) по смешной стоимости. Однако, несмотря на громкое название – Switch, это устройство к коммутатору никакого отношения не имеет. Данное устройство представляет собой простой повторитель, способный работать на одной из двух указанных скоростей – 10 или 100 Мбит/с. Многие не очень грамотные пользователи и горе-администраторы на этом попались, думая, что приобретают полноценный коммутатор.

6.4. Дополнительные функции коммутаторов

Каждый коммутатор увеличивает задержку в сети. Однако эта задержка зависит от типа коммутатора и используемого метода коммутации. В современных коммутаторах используются следующие два режима:

1. Пересылка данных с промежуточным хранением заключается в том, что коммутатор сначала целиком получает пакет, а затем в соответствии с таблицей коммутации пересылает этот пакет получателю. Такая процедура вносит дополнительную задержку в процесс передачи пакета, однако в ожидании полного получения пакета коммутатор может проверить его на целостность данных и наличие в нём ошибок.
2. При использовании сквозной коммутации коммутатор считывает адрес получателя и сразу начинает его передачу получателю. Такой режим передачи ещё называют передачей на лету. Он обеспечивает минимальную задержку.

Метод сквозной коммутации имеет несколько существенных недостатков. При возникновении конфликта каждый узел, обнаруживший конфликт, посылает специальную последовательность, усиливающую конфликт. Коммутатор, реализующий алгоритм коммутации на лету, не в состоянии такую последовательность отфильтровать. Он вынужден её передать. В связи с тем, что длина такой последовательности менее 64 байт, то фильтрация её при другом способе передачи не вызывает проблем. Если коммутатор, получив адресную часть пакета, сразу обращается к коммутационной системе, при этом не сохраняя адресную часть, то в случае невозможности предоставления внутреннего канала передачи, например в связи с занятостью приёмника, он может сохранить только информационную часть пакета, а не весь пакет. Поэтому пакет будет считаться утраченным. Во избежание утраты пакета, коммутатор вынужден запоминать пакет, а если ему удаётся его передать на лету, то пакет из буферного запоминающего устройства удаляется.

Для временного хранения пакетов и последующей их отправки по нужному адресу коммутатор использует буферное запоминающее устройство, которое может быть общим или индивидуальным для каждого порта.

При наличии индивидуальных запоминающих устройств для каждого порта пакеты хранятся в очередях, количество которых соответствует количеству выходных портов. Пакет передается на выходной порт только тогда, когда все пакеты, находившиеся впереди него в очереди, были успешно переданы.

При использовании общей памяти все пакеты хранятся в общем буфере,

который используется всеми портами коммутатора. Количество памяти, отводимой порту, определяется требуемым ему количеством. Такой метод называется динамическим распределением буферной памяти. После этого пакеты, находящиеся в буфере, динамически распределяются по выходным портам. Это позволяет получить пакет с одного порта и отправить его на другой порт, не создавая очередей.

Каждый из подходов имеет свои достоинства и недостатки, поэтому выбор того или иного решения зависит только от разработчика коммутатора.

Поскольку коммутатор представляет собой сложное вычислительное устройство, имеющее несколько процессорных модулей, то, естественно, целесообразно нагрузить их помимо выполнения основной функции передачи кадров с порта на порт по алгоритму моста и некоторыми дополнительными функциями, полезными при построении надёжных и гибких сетей. Ниже описываются наиболее распространенные дополнительные функции коммутаторов, которые поддерживаются большинством производителей коммуникационного оборудования.

6.4.1. Идентификация источника

Процессор порта коммутатора может запоминать MAC-адрес подключённого к нему узла или адреса нескольких узлов, если они подключены через повторитель (концентратор). Это позволяет защитить сеть от несанкционированного подключения. Администратор сети имеет возможность определять время хранения MAC-адреса, устанавливать MAC-адреса, изменять MAC-адреса самостоятельно или динамически.

На рис. 6.13 приведён пример использования метода запоминания MAC-адресов. Один порт хранит целую таблицу MAC-адресов, а другой порт всего один MAC-адрес. Подключенный к этому порту узел имеет MAC-адрес отличный от хранящегося в памяти порта. Поэтому в доступе этому узлу к сети будет отказано. Данный метод проверки адресов работает на аппаратном уровне, поэтому единственный способ обойти такую защиту состоит в генерации узлом нужного MAC-адреса, а это практически невозможно, поскольку необходимо знать, какой адрес нужно генерировать, а самое главное, обойти все «препоны», устанавливаемые сетевой картой узла (адрес аппаратным образом зашит в схемы сетевой карты производителем).

Многие коммутаторы позволяют администраторам задавать дополнительные условия фильтрации кадров наряду со стандартными условиями их фильтрации в соответствии с информацией адресной таблицы. Пользовательские фильтры предназначены для создания дополнительных барьеров на пути пакетов, которые ограничивают доступ определённых групп

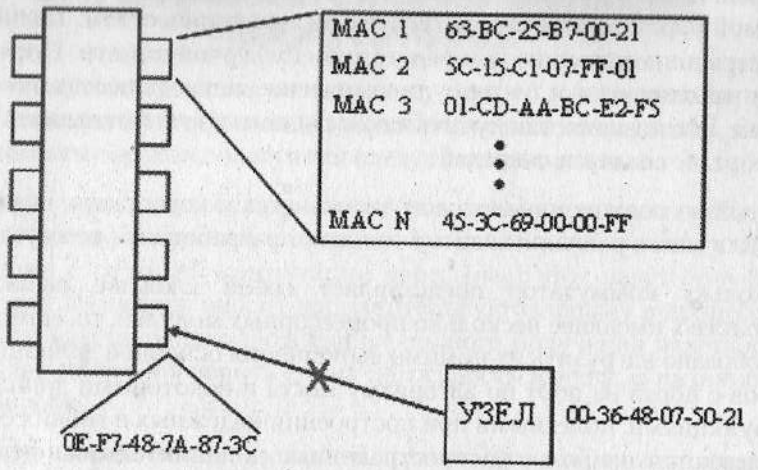


Рис. 6.13. Идентификация источника

пользователей к определённым службам или ресурсам сети.

Наиболее простыми являются пользовательские фильтры на основе физических адресов узлов. Поскольку эти адреса – это та информация, с которой работает коммутатор, то он позволяет задавать такие фильтры в удобной для администратора форме, возможно, проставляя некоторые условия в дополнительном поле адресной таблицы, например, уничтожать пакеты с определённым адресом. При этом пользователю, работающему на компьютере с данным адресом, полностью запрещается доступ к ресурсам сети.

Часто администратору требуется задавать более тонкие условия фильтрации, например, запретить некоторому пользователю печатать свои документы на определённом сервере печати чужого сегмента, а остальные ресурсы этого сегмента сделать доступными. Для реализации такого фильтра нужно запретить передачу пакетов с определённым адресом назначения. Обычно условия фильтрации записываются в виде булевых выражений, формируемых с помощью логических операторов AND и OR.

6.4.2. Подавление источника

Поскольку к одному порту коммутатора подключён только один узел, то он монопольно владеет каналом передачи информации и, естественно, никаких конфликтов за владение каналом связи не возникает. Для временного хранения пакетов коммутатор имеет буферное запоминающее устройство. Если выходной порт коммутатора занят (рис. 6.14), а источник информации

постоянно передаёт пакеты, то в большинстве случаев даже очень большой объём буферной памяти не спасает – она обязательно переполнится. В случае переполнения буфера коммутатор не в состоянии запоминать поступающие пакеты, поэтому они будут утрачены. Для исключения возможности переполнения буфера коммутатор должен иметь некоторый механизм (механизмы), позволяющий ограничивать поток поступающей информации вплоть до полной остановки передачи информации узлом. В общем случае такие механизмы принято называть – подавлением источника. Так как возможны два режима передачи: полудуплексный и полнодуплексный, то рассмотрим механизмы подавления источника более подробно для каждого из режимов.

Полудуплексный режим. В полудуплексном режиме между портом коммутатора и узлом имеется разделяемая среда передачи данных, поэтому единственным способом ограничения скорости передачи информации является создание конфликтной ситуации. Для создания конфликтной ситуации порт коммутатора (рис. 6.14а) нарушает стандартный режим передачи информации. Между пакетами стандартом оговаривается стандартная пауза (9,6 мкс для 10 Мбит Ethernet), при заполнении буфера на 80-95% порт коммутатора посылает узлу пакет через время меньшее стандартной паузы (через 9,5 мкс для 10 Мбит Ethernet). Поскольку узел один, то он через стандартную паузу начинает передачу очередного пакета, и на линии происходит столкновение пакетов. После чего узел отключается на время, выбираемое случайным образом из некоторого диапазона (см. главу 5). Если после конфликта узел опять начинает передачу информации, то коммутатор во время передачи пакета узлом посылает свой пакет, создавая ещё раз конфликт. Поскольку при возникновении каждой последующей конфликтной ситуации время паузы увеличивается, то порт имеет некоторый запас по времени для освобождения буфера. Однако созданием конфликтных ситуаций злоупотреблять нельзя, поскольку после 16-й конфликтной ситуации узел начнёт думать, что произошла какая-то авария с портом или линией, и вообще перестанет передавать информацию, поэтому порт должен следить за количеством созданных им конфликтов. Если буфер не освобождается, то порт должен после 15-й конфликтной ситуации обязательно принять пакет, а затем можно опять создавать конфликты. Для таких случаев в буфере должно быть место для хранения пакетов, поэтому порт всегда резервирует от 5 до 15% объёма буфера. Таким простым способом коммутатор может регулировать темп поступающей информации от узла.

Построение сетей на основе коммутаторов позволяет учитывать степень важности пакетов. Эта возможность является следствием того, что коммутаторы буферизуют кадры перед их отправкой на другой порт. Коммутатор обычно ведёт для каждого входного и выходного порта не одну, а несколько

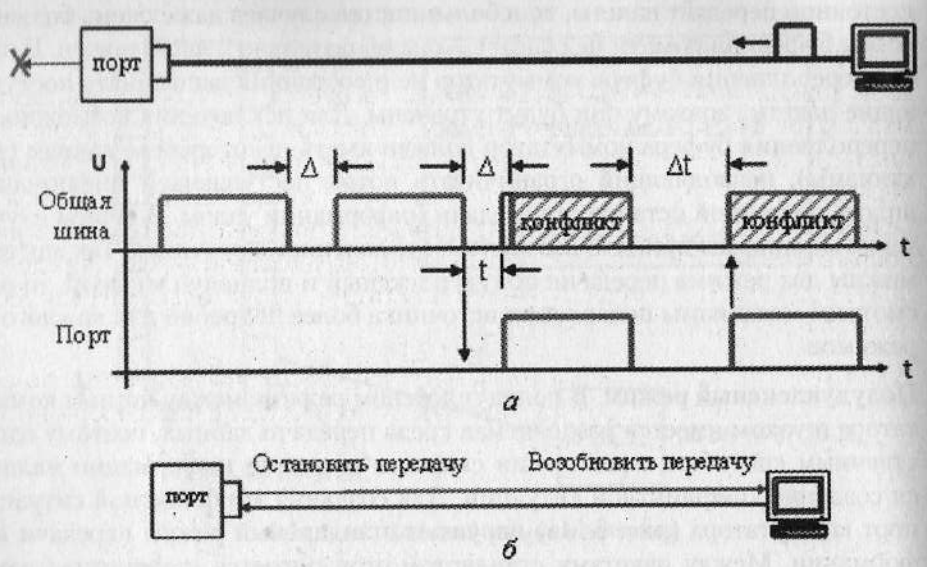


Рис. 6.14. Подавление источника при режимах работы: а) полудуплексном; б) полнодуплексном

очередей, причём каждая очередь имеет свой приоритет обработки. В этом случае, при большем темпе поступления высокоприоритетных пакетов, время ожидания низкоприоритетных пакетов не прогнозируемо и может достигать очень больших величин, что принципиально недопустимо. Поэтому коммутатор должен быть сконфигурирован таким образом, чтобы гарантировать передачу всех пакетов за приемлемое время, например, передавать один низкоприоритетный пакет на каждые 10 высокоприоритетных пакетов.

Поддержка приоритетной обработки особенно важна для приложений, предъявляющих различные требования к допустимым задержкам пакетов и к пропускной способности сети, например IP-телефонии, видео.

Однако к каким уровням задержек приводит назначение того или иного уровня приоритета пакету, какую пропускную способность обеспечивает приоритет потоку пакетов, можно выяснить только путём проведения натурных экспериментов или же с помощью имитационного моделирования.

Полнодуплексный режим. В полнодуплексном режиме между портом коммутатора и узлом имеются два канала передачи данных (рис. 6.14б), по одному в каждую сторону. Порт коммутатора имеет возможность в направлении узла, по свободному каналу связи, передавать служебную информацию для управления узлом. Поэтому при заполнении буфера порт коммутатора узлу передаёт команду «остановить передачу», а как только буфер освобождается

на определённую величину, порт коммутатора узлу передаёт команду «возобновить передачу».

Для нормальной работы коммутатора требуется отсутствие замкнутых маршрутов в сети. Эти маршруты могут создаваться администратором специально для образования резервных связей или же возникать случайным образом, что вполне возможно, если сеть имеет многочисленные связи, а кабельная система плохо структурирована или документирована. Алгоритм покрывающего дерева позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой.

В связи с тем, что механизм создания связующего дерева достаточно сложен и поддерживается соответствующим протоколом, то целесообразно этому вопросу посвятить целую главу.

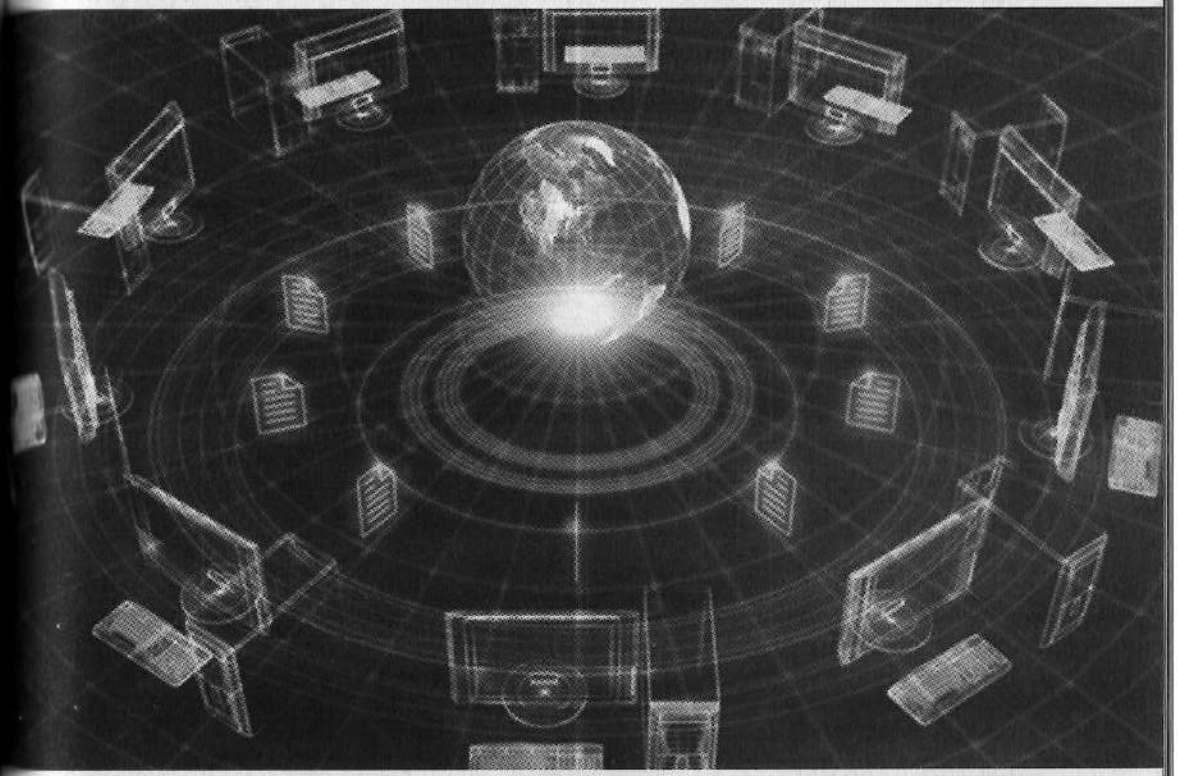
Контрольные вопросы

1. *Почему при росте числа узлов в сети производительность последней снижается?*
2. *Какие существуют методы уменьшения числа конфликтов в сети?*
3. *Дайте определение моста.*
4. *Каким образом мост делит сеть на части?*
5. *Почему при использовании моста производительность сети возрастает?*
6. *Почему мосты нельзя использовать в петлевых структурах?*
7. *Как называется вырожденная сеть, построенная на мостах?*
8. *Какое название получил мультимост?*
9. *Назовите основные технологии построения коммутирующей системы коммутатора.*
10. *Назовите достоинства и недостатки коммутирующей системы на основе общей шины.*
11. *Назовите достоинства и недостатки коммутирующей системы на основе коммутационной матрицы.*
12. *Назовите достоинства и недостатки коммутирующей системы на основе многовходовой разделяемой памяти.*
13. *За счёт чего коммутаторы увеличивают производительность сети?*
14. *Почему использование коммутатора в сетях с сервером не даёт никакого выигрыша?*

15. Почему в коммутаторах используют порты разной производительности?
16. Чем вызвана задержка в коммутаторах?
17. Назовите достоинства и недостатки коммутаторов, выполненных в виде законченного устройства.
18. Чем обусловлена необходимость в появлении коммутаторов стекового типа?
19. Назовите достоинства и недостатки коммутаторов, выполненных в виде модульного устройства.
20. Назовите известные Вам скорости передачи данных в коммутаторах.
21. Назовите дополнительные функции коммутаторов.
22. Каким образом осуществляется идентификация источника коммутатором?
23. Каким образом коммутатор управляет темпом поступления информации от источника?

Глава 7.

СВЯЗУЮЩЕЕ ДЕРЕВО



Наличие избыточности в сети является важным фактором. Избыточность делает сеть устойчивой к возможным сбоям. Избыточность топологии защищает сеть и обеспечивает её функционирование в случае сбоя в отдельном канале, на порте или в сетевом устройстве. Однако топологии сетей, построенных на использовании коммутаторов, обладающие избыточностью, подвержены широковещательным штормам, множественным передачам кадров и неустойчивости баз данных, в которых находятся MAC-адреса управления доступом к среде передачи.

В настоящее время частные компании и государственные организации в своей деятельности все более полагаются на компьютерные сети. В случае неработоспособности сети резко падает производительность работы компании, а потребители выражают неудовлетворенность. Поэтому компании стремятся добиться круглосуточной работоспособности сети на протяжении всей рабочей недели.

Достижение 100% работоспособности сети невозможно, однако обеспечение работоспособности сети на 99,999% (такой уровень надежности называется «пятью девятками») – цель, которую ставят перед собой компании и организации. Это значение можно представить в среднем как один день простоя сети за 30 лет, один час простоя за 4000 дней или 5,25 минут простоя за год. Эта оценка применима и к другим важным службам, таким как финансовые операции, больницы, работа провайдеров и т.д. Такая надёжность достигается за счёт использования надёжного оборудования и путём проектирования сети, устойчивой к сбоям и ошибкам. Для этого сеть проектируется таким образом, чтобы в ней быстро происходила реконфигурация сети, позволяющая обойти участок со сбоем.

Устойчивость к сбоям достигается за счёт создания избыточности в сети. Избыточность означает наличие в сети средств, которые превышают требования обычной ситуации при работе сети. Целью создания избыточности в сети является предотвращение простоя в сети в тех случаях, когда в отдельной точке сети происходит сбой. Для повышения надёжности все сети должны обладать избыточностью. Примером сети с избыточностью может служить сеть автомобильных дорог. Если одна трасса закрыта для ремонта, то имеются другие маршруты к требуемому месту.

Как было сказано ранее, коммутаторы в сетях, содержащих петли, не функционируют. Протокол связующего дерева позволяет решить эту проблему и переводит все резервные пути данных в заблокированное состояние, создавая только один путь через сеть. Если любой из путей пересылки через один из сегментов сети в связующем дереве стал недостижимым, протокол связующего перестраивает топологию связующего дерева и использует ранее

заблокированные связи в сети, он определяет топологию сети, используя алгоритм связующего дерева.

Чтобы избежать путаницы, следует заметить, что **протокол связующего дерева (STP)** и **алгоритм связующего дерева (STA)** – два отдельных объекта. Алгоритм описывает последовательность действий, которая приводит к выявлению петель и отключению резервных связей. Он выбирает контрольную точку в сети и определяет резервные пути к этой контрольной точке. Если он находит резервные пути, то выбирает один из путей для пересылки, а остальные блокирует. Протокол же связующего дерева обеспечивает передачу всей информации о состоянии сети, необходимой для построения связующего дерева и ликвидации всех резервных связей в пределах сети.

Алгоритм связующего дерева использует решения, полученные в соответствии с теорией графов, также известные как алгоритм самого короткого пути для вычисления маршрутов, развитый Э. Дэйкстрой, чтобы строить подмножество топологии сети без петель.

Теория заявляет, что для связанного графа, состоящего из интерфейсов (вершин) и граней, соединяющих пары интерфейсов, связующее дерево для граней обеспечивает связность графа, при этом не содержащего петли.

Алгоритм строит направленный граф, где каждая связь представлена вершинами и «взвешенными» гранями, как показано на рис. 7.1.

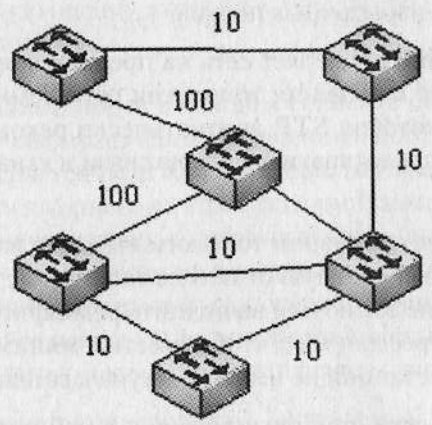


Рис. 7.1. Сеть, содержащая петли

Каждая связь имеет стоимость. Взвешенные грани, которые обычно имеют большее количество переходов, чем прямые связи, имеют более высокие значения. Сумма значений до данной точки или получателя составляет

полное взвешенное значение пути. Минимальное полное взвешенное значение означает наиболее эффективный путь от одной точки до другой.

Алгоритм связующего дерева назначает каждому коммутатору сети уникальный идентификатор и присваивает приоритет. Затем он назначает уникальный идентификатор каждому порту коммутатора, который обычно представляет собой собственный MAC-адрес интерфейса порта. Каждому порту моста ставится в соответствие стоимость пути, которая назначается по умолчанию или вручную администратором сети и представляет собой стоимость передачи кадра в сеть через этот порт.

На основе теории графов сначала выбирается контрольная точка сети, относительно которой вычисляются все возможные пути к этой точке от других точек. После того как все связи найдены, выбирается единственный путь через сеть, а резервные связи блокируются. Контрольная точка в алгоритме связующего дерева называется корневым мостом.

7.1. Работа протокола связующего дерева STP

Протокол связующего дерева является протоколом управления, который используется для поддержки такого состояния сети, в котором в ней отсутствуют петли. Для того чтобы сеть была свободна от петель, при её обнаружении коммутатор автоматически осуществляет логическую блокировку одного или нескольких избыточных портов.

Протокол STP постоянно проверяет сеть на предмет появления нового канала или сбоя. В случае изменения топологии сети коммутаторы, на которых функционирует протокол STP, автоматически реконфигурируют свои порты для того, чтобы предотвратить потерю связи в каналах и возникновение петель.

Если какой-либо элемент активной топологи выходит из строя, то необходимо определить новую свободную от петель топологию. Вычислить новую свободную от петель топологию или выполнить реконфигурацию сети необходимо как можно быстрее для того, чтобы свести к минимуму время, в течение которого конечные станции не имеют доступа к сетевым ресурсам.

После того, как работа сети стабилизирована и в ней произошла реконфигурация, сеть имеет одно связующее дерево. В результате этого для каждой сети, использующей коммутацию, будут выполнены следующие условия:

- в каждой сети существует один корневой мост;
- у каждого моста, который не является корневым, имеется один корневой порт;

- в каждом сегменте имеется один назначенный порт;
- порты, которые не являются назначенными, не используются.

Для пересылки кадров используются только корневые и назначенные порты. Порты, которые не являются назначенными, отбрасывают поступающие на них кадры. Такие порты называются блокирующими или отбрасывающими. Корневым портом коммутатора называется порт, ближайший к корневому коммутатору. Для каждого некорневого коммутатора должен быть выбран один корневой порт. Алгоритм STA заключается в выполнении следующих шагов:

1. **Выбор корневого коммутатора.** В одной конкретной сети в качестве корневого коммутатора может выступать только один коммутатор. У корневого коммутатора все порты являются назначенными. Назначенные порты в обычном режиме осуществляют пересылку кадров. Когда порт находится в таком состоянии, он может получать и отправлять потоки данных.
2. **Выбор корневого порта на некорневых коммутаторах.** Для каждого некорневого коммутатора протокол STP устанавливает один корневой порт. В качестве корневого порта выбирается маршрут от некорневого коммутатора до корневого коммутатора с наименьшей оценкой. Корневые порты в обычном состоянии осуществляют пересылку кадров. Оценка маршрута в алгоритме связующего дерева представляет собой аккумулированную оценку, которая вычисляется на основе ширины полосы пропускания.
3. **Выбор назначенного порта в каждом сегменте сети**. В каждом сегменте протокол STP выбирает один назначенный порт. В качестве назначенного порта выбирается порт коммутатора, который имеет наименьшую оценку маршрута к корневому коммутатору. Назначенные порты в обычном состоянии осуществляют пересылку кадров для данного сегмента. Порты, которые не являются назначенными, в обычной ситуации находятся в состоянии блокировки для осуществления логического разбиения топологии, содержащей петли. Когда порт находится в состоянии блокировки, он не пересылает кадры, однако может их получать.

Протокол STP выбирает корневой узел, называемый корневым коммутатором, а затем строит топологию сети, в которой к каждому сетевому узлу существует только один маршрут. Образующееся при этом дерево исходит из корневого коммутатора. Избыточные каналы, которые не являются частью дерева кратчайших маршрутов, блокируются. Создание свободной от петель топологии становится возможным именно из-за того, что некоторые

маршруты блокируются. Пакеты данных, поступающие в заблокированный канал, отбрасываются.

Протокол STP требует, чтобы сетевые устройства обменивались сообщениями для обнаружения петель. Каналы, которые вызывают появление петли, переводятся в заблокированное состояние. Сообщения, которые посылают коммутаторы для того, чтобы можно было создать свободную от петель логическую топологию, называются модулями данных мостового протокола BPDU. Эти модули BPDU продолжают приниматься на заблокированных портах, что обеспечивает возможность построения нового связующего дерева в случае, если происходит сбой в устройстве или на активном маршруте.

BPDU – это сообщения коммутаторов, при помощи которых STP вычисляет и изучает топологию сети. Эти сообщения содержат информацию, которая идентифицирует канал связи, предполагаемый корневой коммутатор, расчётную стоимость корневого пути, другие коммутаторы STP, идентификаторы порта и время жизни сообщения.

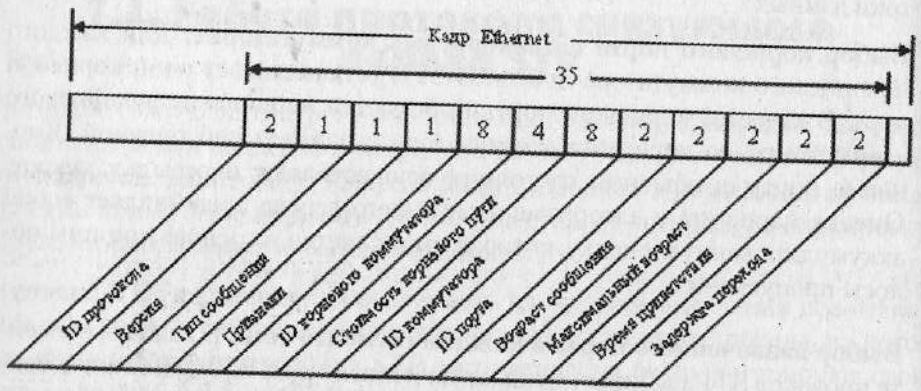


Рис. 7.2. Формат сообщения BPDU

Коммутаторы STP обмениваются сообщениями BPDU через определённый интервал времени, обычно несколько секунд. По умолчанию BPDU посылаются из всех портов каждые две секунды, чтобы гарантировать устойчивую сеть без случайных петель данных. Если один или более коммутаторов выходят из строя или к сети присоединяется другой член STP, сообщения BPDU помогут другим коммутаторам заметить это изменение в топологии сети. Соседние коммутаторы обнаружат отсутствие сообщений конфигурации и заново начнут вычисления по построению связующего дерева.

Члены связующего дерева никогда не пересылают сообщения BPDU. Вместо этого каждый коммутатор принимает локальное решение на основании полученных сообщений BPDU, далее он создает новое сообщение BPDU на основании новой информации и посылает это сообщение из своих портов.

Формат модуля данных мостового протокола BPDU приведён на рис. 7.2, он содержит 35 байт. Модуль данных мостового протокола BPDU содержит следующие поля:

- **ID (идентификатор) протокола** - содержит 2 байта и имеет нулевое значение;
- **Версия** – содержит 1 байт и имеет нулевое значение;
- **Тип сообщения** – содержит 1 байт и имеет нулевое значение;
- **Признаки** – содержит 1 байт, но используются только первые 2 бита. Бит изменения топологии (TCN) сигнализирует о том, что было изменение топологии. Бит подтверждения изменения топологии (ACK) подтверждает получение конфигурационного сообщения с установленным сигнальным битом TCN;
- **ID корневого коммутатора** – содержит 8 байт, которые идентифицируют корневой коммутатор. Первые 2 байта определяют приоритет коммутатора, последующие 6 байт – его идентификатор;
- **Стоимость корневого пути** – содержит 4 байта и характеризует стоимость пути от коммутатора, посылающего сообщение конфигурации, до корневого коммутатора;
- **ID коммутатора** – содержит 8 байт и определяет приоритет и идентификатор коммутатора, посылающего сообщение;
- **ID порта** – содержит 2 байта и определяет порт, из которого было послано сообщение конфигурации. Это поле позволяет STP немедленно обнаруживать петли;
- **Возраст сообщения** – содержит 2 байта и представляет собой количество времени, прошедшее с тех пор, как корневой коммутатор послал сообщение об изменении конфигурации, послужившее основой для данного сообщения;
- **Максимальный возраст** – содержит 2 байта и определяет время действия текущего сообщения конфигурации;
- **Время приветствия** – содержит 2 байта и определяет период времени между сообщениями конфигурации корневого коммутатора;
- **Задержка перехода** – содержит 2 байта и определяет отрезок времени, который коммутатор должен ждать, прежде чем перейти в новое состояние после изменения топологии в сети.

Протокол STP требует, чтобы каждому коммутатору был назначен уникальный 8-ми байтный идентификатор ID. Идентификатор ID состоит из значения

приоритета (2 байта) и MAC-адреса коммутатора (6 байт), как показано на рис. 7.3. Существуют два типа идентификаторов: для использования в системах с виртуальными сетями и без них (о виртуальных сетях речь пойдёт ниже). Значение приоритета в системе без виртуальных сетей назначается из диапазона 0 – 65535. Приоритетом по умолчанию, в соответствии со спецификацией IEEE 802.1d, является значение 32768 (1000 0000 0000 0000 в двоичной записи или 8000 – в шестнадцатеричной). Поле приоритета используется в том случае, если администратор желает повлиять на значение идентификатора путём установки меньшего значения приоритета коммутатора, чем принимаемое по умолчанию. В этом случае ID коммутатора становится меньше.

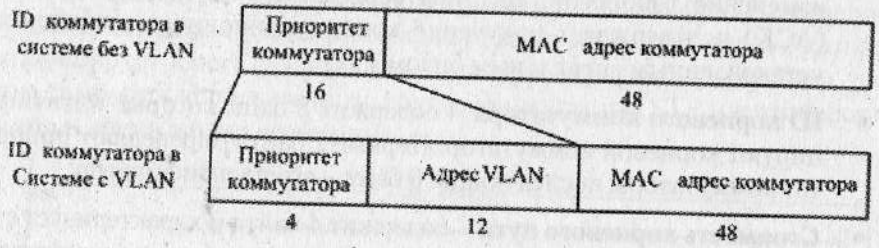


Рис. 7.3. Формат ID коммутатора

В системе с виртуальными сетями возможно всего 16 уровней приоритета, которые должны соответствовать приоритетам системы без виртуальных сетей. Пересчёт приоритетов заключается в том, что приоритеты в системе без виртуальных сетей определяются с ноля с шагом + 4096. В системе с виртуальными сетями суммарный приоритет определяется путём сложения содержимого обоих полей. Например, для виртуальной сети с номером 11 и установленным по умолчанию приоритетом 32768 суммарный приоритет будет иметь значение 32779.

7.2. Таймеры BPDU

В результате задержек распространения модуля данных BPDU в сети он может опаздывать, заставляя коммутатор думать, что произошло изменение топологии сети. Поскольку задержки распространения связаны с разделением среды, длиной пакета, обработкой коммутатором при переходе из порта в порт или любой другой задержкой и могут происходить в любое время, то когда порт коммутатора переходит из блокированного состояния в состояние пересылки, он может создавать временные петли данных, потому что не владеет полной картиной топологии сети (не получил данные об изменении).

Время приветствия определяет временной интервал между регулярными рассылками модулей конфигурации BPDU, генерируемых корневым коммутатором, который зависит от диаметра сети. Под диаметром сети, в данном случае, понимается максимальное количество коммутаторов от корневого до последнего, в самой длинной ветви дерева. Чем больше коммутаторов на пути прохождения кадра, тем больше и задержка. Каждый корневой коммутатор может быть настроен на диаметр сети от 2 до 7 коммутаторов. Максимальное же количество коммутаторов не должно превышать 7. Исходя из всего вышеописанного, время приветствия, определяемое стандартом, находится в пределах 2–20 секунд. По умолчанию принято значение этого интервала, равное 2 секундам.

Остальные коммутаторы только транслируют эти модули BPDU по мере их получения. Иными словами, если в течение 2–20 секунд модули BPDU перестают поступать, то некорневые коммутаторы прекращают рассылать регулярные BPDU. Если этот простой длится более 20 секунд, что является максимальным значением возраста по умолчанию, то коммутатор считает недействительными хранимые им модули BPDU и начинает искать новый корневой порт. Максимальным возрастом считается время, в течение которого мост хранит BPDU перед их уничтожением.

Под задержкой пересылки понимается максимальное время прослушивания (ожидания сообщения BPDU). По умолчанию оно имеет значение 15 секунд, что соответствует максимальному размеру сети в 7 коммутаторов при максимальном количестве утерянных кадров BPDU равном трём и интервале приветствия равном двум секундам. Таймер задержки при пересылке также управляет периодом старения адресной таблицы коммутатора после изменения активной сетевой топологии.

До тех пор, пока коммутатор каждые 2 секунды получает модули BPDU, он поддерживает постоянные копии значений этих модулей. Однако если в устройстве, рассылающем эти модули BPDU с наилучшими маршрутами, происходит сбой, то должен заработать механизм, который позволит другим коммутаторам выполнить функции этого вышедшего из строя устройства. В табл. 7.1 приведены некоторые значения таймеров протокола STP.

Таблица 7.1. *Таймеры протокола STP*

Таймер	Назначение	Значение по умолчанию
Время приветствия	Промежуток времени между регулярными рассылками модулей BPDU корневым коммутатором	2 секунды
Задержка пересылки	Продолжительность прослушивания и изучения топологии	15 секунд

Максимальный возраст	Время хранения модулей BPDU	20 секунд

Следует помнить о двух ключевых моментах, касающихся использования таймеров протокола STP.

Во-первых, не следует изменять значения таймеров, принимаемые по умолчанию, без тщательного анализа. При попытке максимизировать установки таймера следует изменять только значения таймеров на корневом коммутаторе, поскольку модули BPDU содержат три поля, в которых значения таймеров могут быть переданы с корневого моста на все остальные коммутаторы сети.

Во-вторых, изменять таймеры следует путём изменения диаметра сети на корневом коммутаторе, что автоматически приводит к изменению значений таймеров наиболее оптимальным образом.

7.3. Стоимость корневого пути

Оценка маршрута в протоколе связующего дерева представляет собой накопленную общую оценку маршрута, вычисленную на основе полосы пропускания всех каналов на маршруте. В табл. 7.2 приведены оценки маршрутов, определенные в спецификации IEEE 802.1D для различных сетевых технологий, таких как Token Ring, Ethernet и SONET.

Коммутаторы используют понятие оценки для выяснения степени своей близости к другим коммутаторам. Стандарт 802.1D первоначально определял оценку как частное от деления величины 1000 Мбит/с на полосу пропускания канала, выраженную в Мбит/с. С появлением технологии 10 Gigabit Ethernet применение этой формулы стало невозможным. В табл. 7.3 приведены оценки каналов для различных Ethernet-технологий, которые использовались ранее и используются в настоящее время.

Таблица 7.2. *Оценки маршрутов для различных сетевых технологий*

Полоса пропускания	Оценка
4 Мбит/с	250
10 Мбит/с	100
16 Мбит/с	62
45 Мбит/с	39
100 Мбит/с	19
155 Мбит/с	14
622 Мбит/с	6
1 Гбит/	4
10 Гбит/	2

Таблица 7.3. *Пересмотренные оценки каналов для сетей Ethernet*

Скорость канала	Оценка пересмотренная	Оценка предыдущая
10 Гбит/с	2	1
1 Гбит/с	4	1
100 Мбит/с	19	10
10 Мбит/с	100	100

Из таблицы 7.3 видно, как коммутаторы используют оценку маршрута для определения своей близости к другим коммутаторам. Например, канал Ethernet имеет оценку 100 (1000/10), а канал Fast Ethernet имеет оценку, равную 10 (1000/100). Этой схемой пользовался весь мир, однако с появлением технологий Gigabit Ethernet и OC-48 ATM (2.4 Гбит/с) возникла проблема. Например, для OC-48 ATM вычисление оценки по этой формуле даёт значение 1000 Мбит/с / 2400 Мбит/с = 0,41667. Значение 0.41667 недействительно, поскольку его придётся округлить до 1, как и для любой другой технологии с полосой пропускании 1 Гбит/с или более. Единственным выходом из этой ситуации является присвоение оценки 1 всем каналам с полосой пропускания большей или равной 1 Гбит/с. Однако в этом случае протокол STP не сможет выбрать оптимальный маршрут в гигабитных сетях. Для решения этой проблемы IEEE принял решение изменить способ вычисления оценки и использовать нелинейную шкалу.

7.4. Алгоритм работы протокола STP

Когда протокол связующего дерева создает свободную от петель логическую топологию, он всегда использует описанную ниже последовательность действий для принятия решений, состоящую из нескольких этапов.

Определение корневого коммутатора

Когда коммутатор запускается впервые, он считает себя корневым и посылает каждые две секунды модуль BPDU, в котором в качестве ID корневого коммутатора указывает свой идентификатор, который является номером шасси коммутатора (на задней стенке корпуса коммутатора на шильдике указывается номер коммутатора) или номером супервизора (номер управляющей программы, помещённой в коммутатор). Все коммутаторы, получая BPDU, просматривают поле корневого идентификатора, и если в полученном BPDU содержится меньший корневой идентификатор, то происходит пересылка этого модуля другим коммутаторам. Если в поле корневого коммутатора указан больший идентификатор, то такой модуль уничтожается.

Все коммутаторы просматривают поле корневого коммутатора в BPDU, постепенно перестраиваясь таким образом, что выполняют функцию только трансляции, а не генерации модулей BPDU. В конце концов, корневой коммутатор начинает получать модули BPDU со своим идентификатором, которые он будет уничтожать. На этом процесс выбора корневого коммутатора заканчивается. Поскольку максимальная задержка в приёме модулей BPDU составляет 20 секунд, то за это время осуществится выбор корневого коммутатора.

Определение корневых портов

После определения корневого коммутатора все коммутаторы переключаются на определение корневых портов. Корневым портом коммутатора называется порт, ближайший к корневому коммутатору. Каждый некорневой коммутатор должен иметь один корневой порт. Для этого коммутаторы, в качестве меры близости к корневому коммутатору, используют оценку, называемую оценкой корневого маршрута, которая является суммарной оценкой всех каналов, ведущих к корневому коммутатору. На рис. 7.4 приведён пример выбора корневых портов и показано, как вычисляется оценка при проходе через несколько коммутаторов, проиллюстрирован процесс окончательного выбора корневого порта.

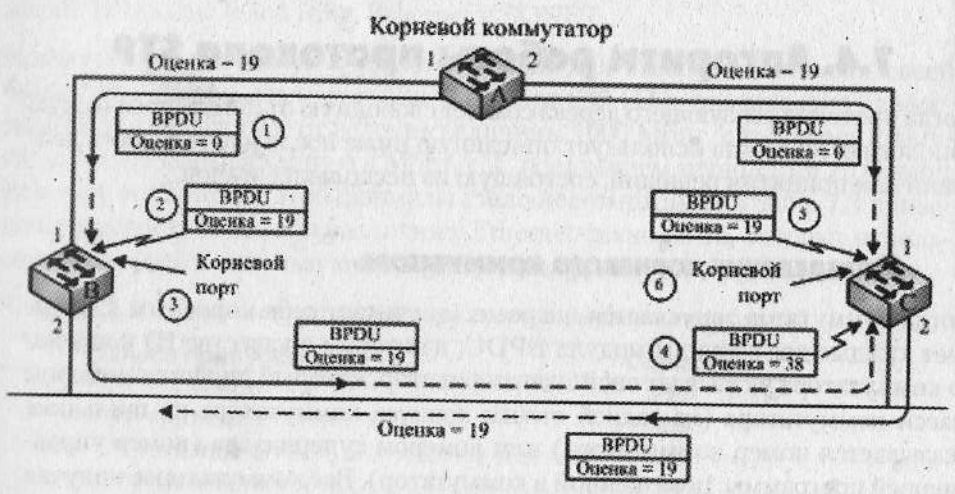


Рис. 7.4. Выбор корневых портов

Допустим, что в качестве корневого коммутатора был выбран коммутатор А. Когда корневой коммутатор А посылает модули BPDU, эти модули содержат оценку корневого маршрута, равную нулю (этап 1). Когда

коммутатор В получает эти модули BPDU, он добавляет оценку маршрута порта А-1 к оценке корневого маршрута, которая содержится в полученном модуле BPDU. Предположим, что сеть функционирует на основе Fast Ethernet. Коммутатор В получает оценку корневого маршрута, равную 0 и добавляет к оценке порта А-1 значение равное 19. После этого коммутатор В использует значение 19 для внутренних расчётов и посылает модули с оценкой корневого маршрута, равной 19, с порта В-2 (этап 3). Когда коммутатор С получает эти модули BPDU от коммутатора В (4-й этап), он увеличивает оценку корневого маршрута до 38 (19 + 19). Однако коммутатор С получает также модули BPDU от корневого коммутатора с порта А-2. Эти BPDU поступают на порт 1 коммутатора С с оценкой 0, и коммутатор С увеличивает внутреннюю оценку до 19 (этап 5). Теперь коммутатору С необходимо принять решение о выборе одного корневого порта – порта, ближайшего к корневому коммутатору. Коммутатор С имеет оценку корневого маршрута равную 19 на порте С-1 и оценку 38 на порте С-2. Следовательно, порт 1 коммутатора С становится корневым портом (этап 6). После этого коммутатор С начинает рассылать оценку корневого маршрута, равную 19 коммутаторам, расположенным в нисходящем направлении (этап 7).

Коммутатор В производит действия аналогичные коммутатору С. В корневой коммутатор с порта 1 коммутатора В можно попасть с оценкой 19, в то время как вычисления для порта 2 коммутатора В дают значение оценки, равное 38. Следовательно, порт В-1 становится корневым портом коммутатора В.

Следует помнить о том, что оценки протокола STP увеличиваются при поступлении модулей BPDU на порт, а не при рассылке их с порта. Например, модули BPDU поступают на порт 1 коммутатора В с оценкой равной 0 и эта оценка увеличивается до 19 "внутри" коммутатора В.

Определение назначенных портов

В каждом сегменте должен быть всего один назначенный порт. Этот порт функционирует как единственный порт коммутатора, который получает и отправляет все данные, поступающие в сегмент и отправляемые из него. Такой подход основывается на том, что если только один порт обрабатывает данные для всех каналов, то все петли оказываются разорванными. Коммутатор, которому принадлежит назначенный порт данного сегмента, называется назначенным коммутатором этого сегмента.

Как и при выборе корневого порта, назначенные порты выбираются на основе обобщённой оценки корневого маршрута к корневому мосту, как показано на рис. 7.5. Рассмотрим назначенные порты и каждый сегмент по

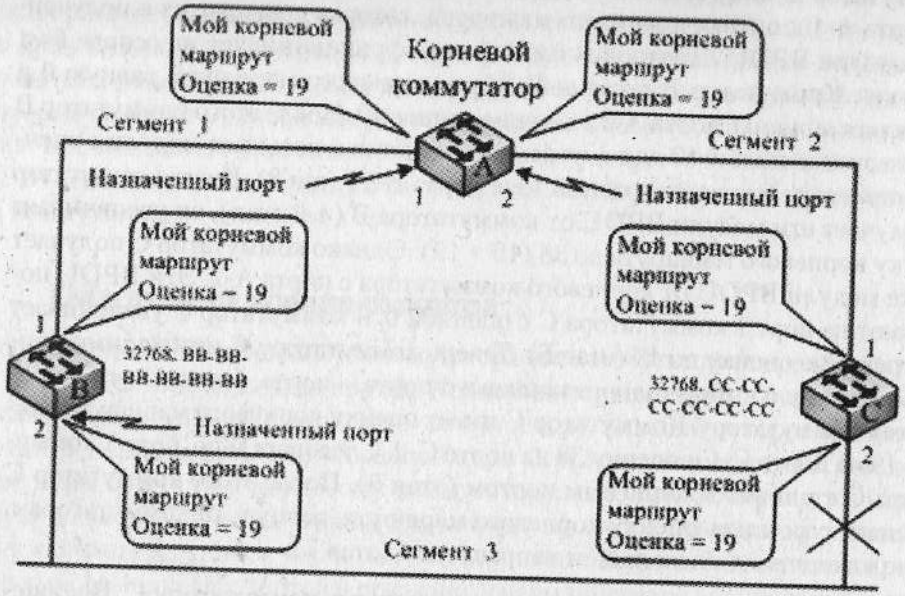


Рис. 7.5. Выбор назначенных портов

очереди. Сначала рассмотрим сегмент 1, который является каналом между коммутаторами A и B. В этом сегменте имеется два порта коммутаторов: порт 1 коммутатора A и порт 1 коммутатора B. Порт 1 коммутатора A имеет оценку корневого маршрута равную 0 (всё-таки это корневой коммутатор), а порт 1 коммутатора B имеет оценку корневого маршрута равную 19 (значение 0 получено в модулях BPDU от коммутатора A плюс оценка маршрута, равная 19, назначенная порту 1 коммутатора B). Поскольку порт 1 коммутатора A имеет меньшую оценку корневого маршрута, он становится назначенным портом сегмента.

Для сегмента 2 (канал от коммутатора A к коммутатору C) имеет место аналогичная операция выбора. Порт 2 коммутатора A имеет оценку корневого маршрута, равную 0, в то время как порт 1 коммутатора C имеет оценку корневого маршрута, равную 19. Порт 2 коммутатора A имеет меньшую оценку и становится назначенным портом. Следует отметить, что каждый активный порт корневого коммутатора становится назначенным портом. Единственным исключением из этого правила является физическая петля на 1-м уровне к корневому коммутатору. Примером является ситуация, в которой два порта корневого коммутатора подсоединены к одному концентратору или когда два порта соединены перекрученным кабелем.

Теперь рассмотрим сегмент 3 (канал от коммутатора B к коммутатору C). Как порт 2 коммутатора B, так и порт 2 коммутатора C имеют оценку корневого маршрута равную 19. В этом случае имеет место равенство оценок. Если протокол STP сталкивается с ситуацией равенства оценок, то при принятии решения он использует четырёхэтапную последовательность:

- выбор наименьшего корневого идентификатора ID;
- вычисление наименьшей оценки маршрута к корневому коммутатору;
- выбор наименьшего идентификатора ID отправителя;
- выбор наименьшего идентификатора ID порта.

В примере, показанном на рис. 7.5, все три коммутатора соглашаются с тем, что коммутатор A является корневым коммутатором. Однако оба коммутатора B и C имеют одинаковые оценки, равные 19. В этом случае решающим становится третий критерий – значение идентификатора ID. Поскольку идентификатор ID коммутатора B (32768. BB-BB-BB-BB-BB-BB) меньше чем идентификатор ID коммутатора C (32768. CC-CC-CC-CC-CC-CC), то порт 2 коммутатора B становится назначенным портом сегмента 3. Соответственно, порт 2 коммутатора C становится не назначенным портом.

Коммутатор использует эту состоящую из четырёх этапов последовательность принятия решений для того, чтобы сохранить копию «наилучшего» модуля BPDU на каждом порту. Когда коммутатор выполняет эту оценку маршрутов, он рассматривает все модули BPDU, которые были получены на данный порт, а также BPDU, которые были бы посланы на этот порт. При поступлении каждого модуля BPDU для него выполняется описанная выше четырёхэтапная проверка с целью выяснить, не имеет ли он меньшее значение оценки. Если новый BPDU или локально сгенерированный BPDU имеет меньшее значение оценки, то он заменяет прежнее значение.

Когда коммутатор в первый раз становится активным, все его порты посылают модули BPDU каждые 2 секунды (если используются значения таймеров по умолчанию). Если от соседнего устройства в течение определенного времени (по умолчанию 20 секунд) не поступают модули BPDU, то локальный порт возобновляет рассылку.

Последовательность состояний порта в протоколе STP

При включении коммутатора он считает себя корневым и переходит в состояние прослушивания. Отсутствие модулей BPDU в течение определённого периода времени (максимальный возраст) может вызвать переход из состояния блокировки в состояние прослушивания. Когда порт находится

в промежуточном состоянии прослушивания, он может отправлять и получать BPDU для определения текущей топологии. В этом состоянии передача данных пользователей не происходит. Именно в состоянии прослушивания коммутатор выполняет операции по выбору корневого коммутатора, корневых портов на некорневых коммутаторах и выбора назначенных портов для каждого сегмента. Порты, которые остаются назначенными или корневыми в течение 15 секунд (задержка пересылки), переходят в состояние изучения топологии. Порты, которые не являются назначенными или корневыми, возвращаются в состояние блокировки.

При переходе порта в состояние изучения топологии он может заполнять свою таблицу MAC-адресов адресами, полученными с его портов, однако не пересылает кадры пользователя. По умолчанию состояние изучения продолжается 15 секунд. В это время коммутатор всё ещё не передаёт данных пользователя. На рис. 7.6 показаны состояния порта, с которыми работает протокол связующего дерева.

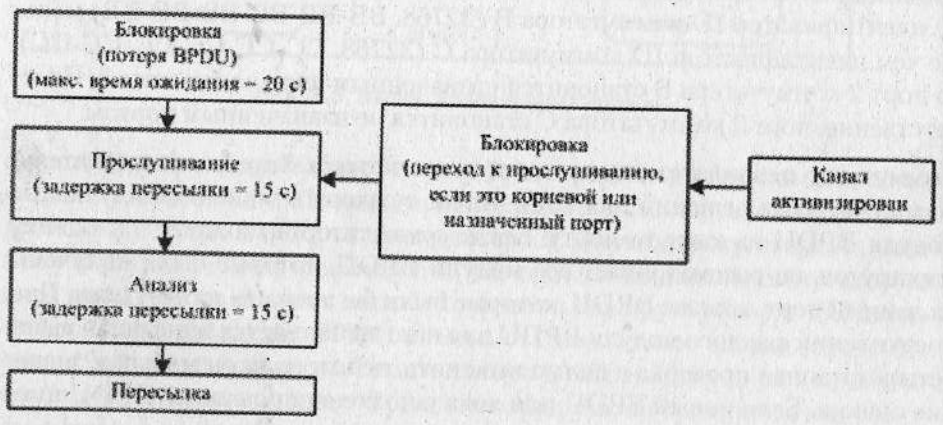


Рис. 7.6. Состояние порта

В состоянии изучения топологии уменьшается объём лавинной рассылки, которая требуется в начале пересылки. Если какой-либо порт по-прежнему является назначенным или корневым портом в заключительной фазе состояния изучения топологии, то он переходит в состояние пересылки. Порты, которые не являются назначенными или корневыми, возвращаются в состояние блокировки. В состоянии пересылки порт может отправлять и получать данные пользователей. Ниже приводятся состояния протокола STP:

- Состояние блокировки. Кадры пользователей не пересылаются, прослушиваются модули BPDU.

- Состояние прослушивания. Кадры пользователей не пересылаются, но прослушиваются.
- Состояние изучения топологии. Кадры пользователей не пересылаются, изучаются адреса других устройств.
- Состояние пересылки. Пересылаются кадры пользователей, и изучаются адреса других устройств.
- Состояние отключения. Кадры пользователей не пересылаются, модули BPDU не прослушиваются.

Обычно время, требуемое порту для перехода из состояния блокировки в состояние пересылки, составляет от 30 до 50 секунд. Это время таймеров протокола связующего дерева может быть изменено сетевым администратором. Обычно эти периоды таймеров устанавливаются со стандартным значением. Значения по умолчанию устанавливаются для того, чтобы у коммутаторов сети была возможность собрать правильную информацию о сетевой топологии.

Время, требуемое порту для перехода из состояния прослушивания в состояние изучения или из состояния изучения топологии в состояние пересылки, называется задержкой пересылки и по умолчанию составляет 15 секунд. Максимальный возраст, который по умолчанию равен 20 секундам, равен времени хранения коммутатором модуля BPDU перед его уничтожением. Если порт находится в состоянии блокировки и не получает новых модулей BPDU за время максимального возраста, то он переходит из состояния блокировки в состояние прослушивания.

Если порт коммутатора подсоединён только к узлам пользователей (не имеет соединений с другими коммутаторами), то на таких портах пользователей следует включить функцию коммутатора «быстрый порт». При включенной функции быстрого порта в момент активизации порта он автоматически переходит из состояния блокировки в состояние пересылки, поскольку через эти порты не могут возникнуть петли.

Перерасчет связей связующего дерева

Если в топологии сети происходят изменения, вызванные сбоем коммутатора или канала, то протокол STP перестраивает соединения в сети путём перевода некоторых заблокированных портов в состояние пересылки. В сети, показанной на рис. 7.7, корневой коммутатор X вышел из строя и соответственно вышли из строя его оба порта. Он не посылает коммутаторам Y и Z модулей BPDU в течение времени максимального возраста (по умолчанию 20 секунд, что равно 10 пропущенным рассылкам модулей BPDU).

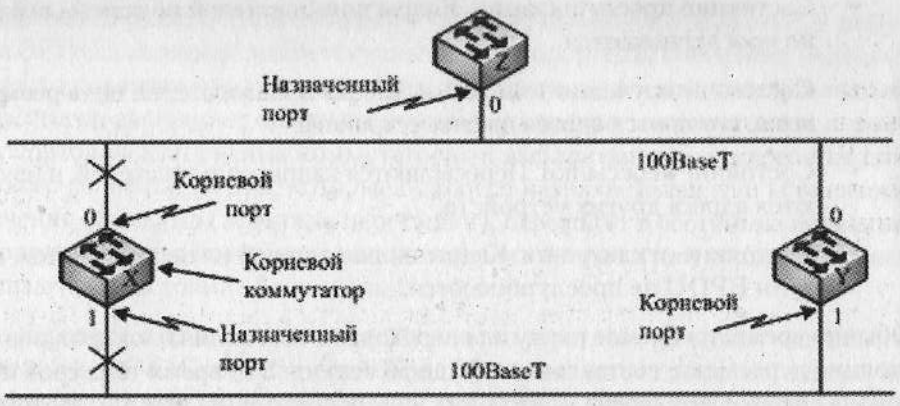


Рис. 7.7. Перерасчет связующего дерева

Отсутствие модулей BPDU обнаруживают оба коммутатора Y и Z. В момент, когда истекает время таймера максимального возраста на коммутаторе Y, а новый BPDU от коммутатора X не поступил, начинается перерасчёт связующего дерева. Коммутатор Y переводит свой заблокированный порт (порт 1) из состояния блокировки в состояние прослушивания, а затем в состояние пересылки. Примерно через 50 секунд, в соответствии с алгоритмом протокола STP, коммутатор Y становится корневым коммутатором и начинает пересылать данные между двумя сегментами.

Преобразование связующего дерева необходимо для нормальной работы сети. Для сети, в которой используются коммутаторы, ключевым вопросом является время, которое требуется для расчёта нового связующего дерева в случае изменения её топологии. Быстрое преобразование является желательным, поскольку в этом случае уменьшается время нахождения портов коммутаторов в переходных состояниях. Обычно переход на новое связующее дерево занимает от 30 до 50 секунд.

7.5. Протокол RSTP

Протокол RSTP (быстрый STP) значительно сокращает время перехода с одной топологии связующего дерева на другую, в случае изменения физической топологии или изменения параметров конфигурации. В протоколе RSTP расширены функции, выполняемые альтернативными и резервными портами, а также состояния портов: уничтожения, изучения топологии и пересылки. Протокол RSTP выбирает один коммутатор в качестве корня, связанного со связующим деревом активной топологии, и назначает роли

отдельным портам коммутатора в зависимости от того, являются ли они частью активной топологии.

Новый корневой порт и назначенный порт коммутатора переходят в состояние пересылки путём явного обмена сообщениями между ними. Протокол RSTP позволяет сконфигурировать порты коммутатора таким образом, чтобы они могли перейти в состояние пересылки сразу после повторной инициализации коммутатора. В протоколе RSTP коммутатор может быть:

- корневым, тогда он пересылает кадры;
- назначенным, тогда он пересылает кадры в своих сегментах сети;
- альтернативным, тогда он имеет альтернативный маршрут к корневому коммутатору;
- резервным, тогда он имеет резервный маршрут по отношению к назначенному, ведущий к листьям связующего дерева. Резервные порты могут существовать только в том случае, когда два порта образуют петлю посредством канала типа «точка-точка» или с помощью коммутатора, имеющего два или более соединения с общим сегментом сети;
- отключенным, тогда он не участвует в работе протокола связующего дерева.

Корневой или назначенный коммутатор включает в себя порт в активной топологии. Альтернативный или резервный порт исключается из активной топологии сети.

Состояние порта управляет процессами пересылки и изучения топологии, а также предоставляет значения для удаления, изучения топологии и пересылки. В табл. 7.4 приведены состояния портов в протоколах STP и RSTP.

Таблица 7.4. *Состояния портов в протоколах STP и RSTP*

Состояние порта			Включён в активную топологию?
Операционное	В протоколе STP	В протоколе RSTP	
Включён	Блокировка	Удаление	Нет
Включён	Прослушивание	Удаление	Нет
Включён	Изучение топологии	Изучение топологии	Да
Включён	Пересылка	Пересылка	Да
Отключён	Блокирован	Удаление	Нет

В условиях стабильной топологии протокол RSTP обеспечивает переход каждого корневого порта и назначенного порта в состояние пересылки, в то время как все альтернативные и резервные порты постоянно находятся в состоянии отбрасывания.

Быстрый переход в состояние пересылки является наиболее важной функцией, введённой в протокол RSTP. В протоколе STP реализуемый алгоритм связующего дерева не позволял переход порта в состояние пересылки, пока не закончится перестройка связующего дерева. В новом протоколе RSTP порт может перейти в состояние пересылки, независимо от конфигурации таймера. Для достижения быстрой сходимости дерева протокол RSTP использует две новых переменных – краевой порт и тип канала.

Рис. 7.8. Формат байта признаков сообщения BPDU протокола RSTP

Краевым портом называются порты, непосредственно подсоединенные к конечным узлам. Они не могут создавать петель в сети, следовательно, могут сразу перейти в состояние пересылки, минуя состояния прослушивания и изучения топологии. Краевой порт при включении или выключении не вызывает изменений в топологии сети.

В современных сетях порт, работающий в полнодуплексном режиме, является каналом типа «точка-точка», в то время как порт, функционирующий в полудуплексном режиме, по умолчанию рассматривается как совместно используемый.

Для работы алгоритма связующего дерева протокола RSTP используется байт признаков сообщения BPDU, формат которого приведён на рис. 7.8. Помимо двух основных признаков – изменения топологии и подтверждения изменения топологии, которые сигнализируют об изменении топологии и подтверждении получения конфигурационного сообщения BPDU с установленным признаком изменения топологии, применяются ещё шесть признаков.

Признаки 1,6 сигнализируют о начале и об о окончании процесса. А признаки со 2 по 5 определяют тип и состояние порта, сгенерировавшего сообщение BPDU.

7.6. Протокол агрегирования порта

Всё возрастающие требования пользователей к полосе пропускания и надёжности линий связи между коммутаторами объясняются использованием всё новых приложений, передающих большие объёмы информации. Использование быстродействующих и надёжных каналов связи не всегда оправдано из соображений высокой цены последних, особенно в локальных сетях. Поэтому естественным путём увеличения пропускной способности линии является совместное использование нескольких линий связи между коммутаторами. Однако для их использования они должны быть представлены как единственный канал. Такое объединение возможно на логическом уровне, в результате чего несколько физических связей между коммутаторами рассматриваются как один канал большей производительности. Кроме повышения производительности логический канал увеличивает надёжность связи, поскольку при отказе одной или даже нескольких физических линий логический канал сохраняет работоспособность при одновременном уменьшении пропускной способности.

Для объединения физических линий связи в один логический канал фирма Cisco предложила метод, получивший название EtherChannel. Первоначально этот метод был рассчитан для соединения коммутатора с коммутатором и объединения линий связи Fast и Gigabit Ethernet в один логический канал, что оказалось более дешёвым решением, чем использование более скоростных линий связи. Впоследствии этот метод был распространён на маршрутизаторы и серверы и позволил объединять в один логический канал 2, 4 или 8 линий (портов коммутатора, маршрутизатора, сервера). При использовании линий, работающих в дуплексном режиме, пропускная способность канала увеличивается до 16 раз. Однако, если линии, объединяющиеся в канал, имеют разную производительность и режим работы (полудуплексный или дуплексный), то скорость линий в канале определяется минимальной скоростью линии, а режим работы – полудуплексной линией.

Распределение нагрузки по линиям канала механизм EtherChannel самостоятельно не производит, политика распределения определяется администратором и используемой платформой. Привязка линий в канале может осуществляться на основе MAC-адреса, IP-адреса или на основе протокола UDP, использующего совместно IP+TCP пользовательские пакеты. На рис. 7.9. показан пример создания такого канала.

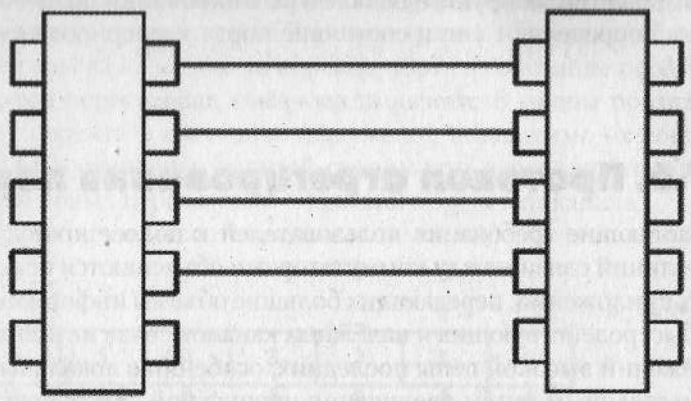


Рис. 7.9. Организация EtherChannel

Для реализации метода EtherChannel можно использовать оригинальный протокол фирмы Cisco – протокол агрегирования порта PAgP или протокол управления агрегированием LACP, определённый стандартом IEEE 802.3ad. Оба протокола используют сообщения для формирования канала. После того как канал организован, он может быть добавлен к связующему дереву в виде единственного порта коммутатора.

Проколы LACP и PAgP посылают каждые 30 секунд сообщения BDUP, используя широковещательный MAC-адрес 01-00-OC-CC-CC-CC с указанием типа протокола (для протокола PAgP приоритет имеет значение 0104). Они проверяют корректность конфигурирования порта и следят за изменениями и отказами связей между двумя коммутаторами. Они гарантируют, что после создания канала EtherChannel все порты будут одинаково сконфигурированы, причём это делается принудительно. Любая модификация порта после создания канала изменит настройки всех других портов канала.

Проколы организации EtherChannel не могут формировать канал на портах, которые настроены для динамических виртуальных сетей VLAN. Виртуальные сети должны быть статическими. Можно также настроить предельные значения для широковещания в процентном выражении или в виде количества пакетов в секунду. Режим количества пакетов в секунду позволяет уничтожать однонаправленные пакеты, когда превышен установленный предел.

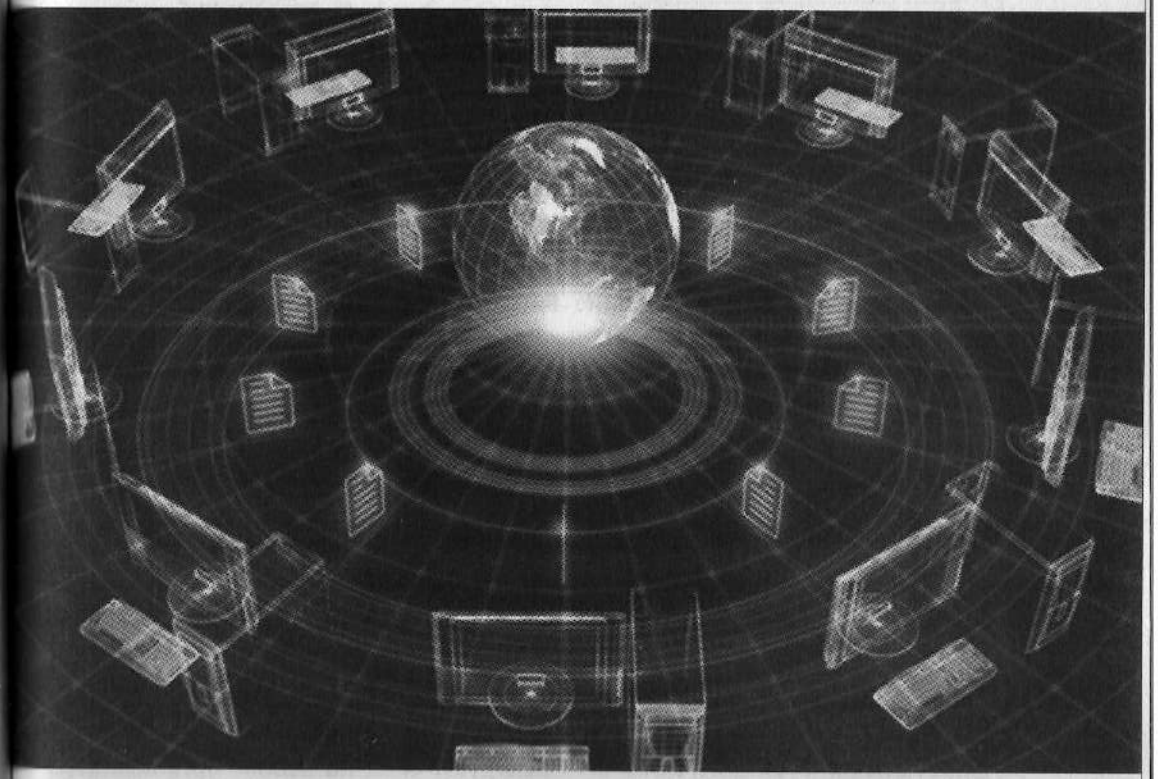
Контрольные вопросы

1. *Поясните назначение протокола связующего дерева.*
2. *Поясните отличия протокола связующего дерева от алгоритма его реализующего.*
3. *Поясните, почему коммутаторы не могут работать в петлевых структурах.*
4. *Поясните назначение корневого коммутатора в алгоритме STP.*
5. *Поясните назначение корневого порта на некорневых коммутаторах в алгоритме STP.*
6. *Поясните предназначение назначенного порта в алгоритме STP.*
7. *Поясните состав сообщения BPDU.*
8. *Как определяется идентификатор коммутатора в протоколе STP?*
9. *Поясните назначение таймеров в алгоритме STP.*
10. *Поясните, в каких единицах измеряется стоимость корневого пути.*
11. *Поясните, для каких целей используется стоимость корневого пути в алгоритме STP.*
12. *Поясните, каким образом определяется корневой коммутатор в алгоритме STP.*
13. *Поясните, каким образом определяется корневой порт в алгоритме STP.*
14. *Поясните, каким образом определяется назначенный порт в алгоритме STP.*
15. *Перечислите состояния порта, которые он принимает в алгоритме STP.*
16. *Поясните, какие действия происходят со связующим деревом после изменения топологии сети.*
17. *Сколько времени необходимо для построения связующего дерева?*
18. *Поясните, каким образом можно сократить время, необходимое для построения связующего дерева.*
19. *Приведите отличия протокола RSTP от STP.*
20. *За счёт чего протокол RSTP имеет меньшее время сходимости?*
21. *Каким образом можно создать связующие деревья в виртуальных сетях?*
22. *Как в протоколе MSTP решается проблема соответствия виртуальных сетей связующим деревьям?*

23. *Поясните термин "зона действия MSTP".*
24. *Как можно увеличить производительность магистральной линии связи?*
25. *В чём заключается действие протокола агрегирования порта?*

Глава 8.

ВИРТУАЛЬНЫЕ ЛОКАЛЬНЫЕ СЕТИ



Большинство сетевых проблем с Ethernet вызвано слишком большим количеством узлов, пробующих послать данные одновременно, вызывая, таким образом, конфликты и перегрузку канала. Столкнувшись с проблемой медленной работы сети, многие администраторы делают одинаковые ошибки – начинают предпринимать меры по модернизации кабельной системы, переходя на более высокие скорости. Применение коммутаторов позволяет резко поднять производительность среды за счёт передачи данных только нужному узлу, а не всем. Но коммутаторы не спасают от широковещательных сообщений. Если широковещательных сообщений передаётся слишком много, то такую ситуацию принято называть – широковещательным штормом.

В компаниях и организациях часто возникает проблема создания рабочих групп. В рабочую группу включается несколько пользователей и сетевых ресурсов – принтеров, сканеров, графопостроителей и т.д. Для качественной работы группы обычно требуется высокая пропускная способность сети, а иногда и конфиденциальность информации. При использовании коммутаторов, не поддерживающих виртуальные сети, для каждой группы необходимо выделить отдельный коммутатор, к которому подключить всех пользователей и необходимые им ресурсы. При большом количестве часто меняющихся по составу рабочих групп необходимо иметь достаточное количество коммутаторов и создание групп производить вручную, методом простой коммутации, что весьма накладно.

Коммутаторы, поддерживающие виртуальные сети, позволяют создавать группы путём программной реализации – конфигурированием коммутатора – и не требуют переконфигурирования физической сети. Слово «виртуальная» обозначает несуществующая или кажущаяся, то есть коммутатор создаёт несуществующую физически сеть. Узлы одной виртуальные сети не имеют возможности передачи данных узлам другой виртуальной сети. Передача данных возможна только с помощью внешних носителей или специальных устройств – маршрутизаторов, речь о которых пойдёт в следующих главах.

Виртуальная сеть представляет собой группу сетевых устройств и служб, не ограниченную физическим сегментом или коммутатором. На рис. 8.1 показано логическое группирование рабочих станций в сети VLAN, в сравнении с физическим группированием рабочих станций в традиционной сети.

Виртуальные сети логически сегментируют сеть, на основе организационных функций узлов, принадлежности к различным рабочим коллективам (группам) или используемым приложениям, а не на базе физического или

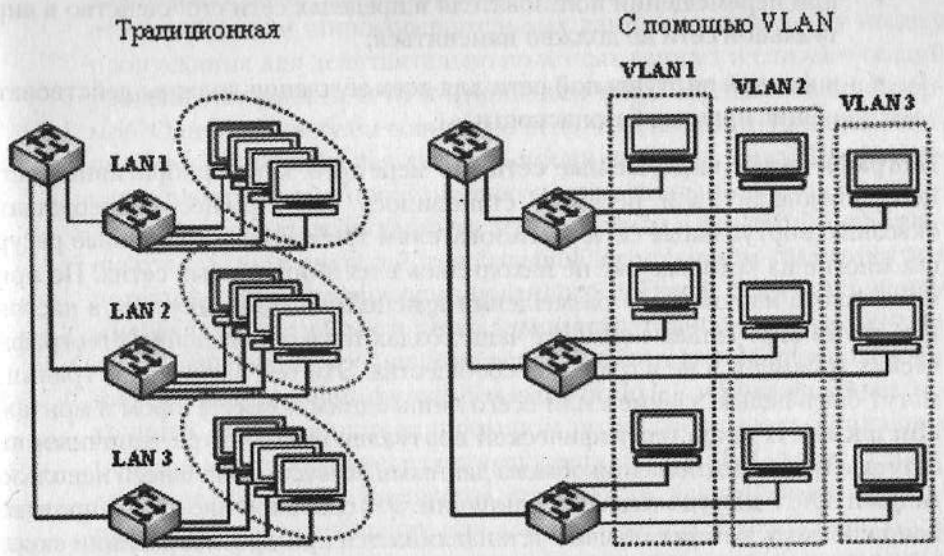


Рис. 8.1. Сети LAN и VLAN и физические границы

географического расположения. Для передачи информации между виртуальными сетями необходимо использовать маршрутизатор. Маршрутизатор устройство достаточно дорогое, поэтому в настоящее время выпускаются коммутаторы, позволяющие осуществлять маршрутизацию между виртуальными сетями.

В технической литературе виртуальные сети подразделяют на сквозные и географические.

Сквозные виртуальные сети позволяют группировать устройства на основе использования ресурсов. Цель сквозных виртуальных сетей состоит в том, чтобы не менее 80% данных передавались внутри локальной виртуальной сети. Сквозные виртуальные сети обладают следующими характеристиками:

- пользователи группируются в виртуальные сети независимо от их физического расположения, но в соответствии с исполняемыми ими рабочими функциями или с принадлежностью к разным рабочим группам;
- у всех пользователей соотношение передачи данных внутри виртуальной сети и за её пределами должно быть одним и тем же и составлять примерно 4:1;

- при перемещении пользователя в пределах сети его членство в виртуальной сети не должно изменяться;
- в каждой виртуальной сети для всех её членов должны действовать общие правила безопасности.

Географические виртуальные сети. По мере того, как корпоративные сети централизовали свои ресурсы, становилось все сложнее поддерживать сквозные виртуальные сети. Пользователям требовались различные ресурсы, многие из которых уже не находились в их виртуальных сетях. По причине такого изменения в размещении и использовании ресурсов в настоящее время виртуальные сети всё чаще создаются в определенных географических границах, а не в границах сообщества. Эти географические границы могут быть целым зданием или всего лишь одним коммутатором в монтажном шкафу. В такой географической виртуальной структуре типичным является случай соотношения обмена данными 20% для локального использования и 80% – для удалённых соединений. Это соотношение прямо противоположно тому, которое обычно устанавливается при проектировании сквозных виртуальных сетей. Хотя такая топология означает, что пользователю приходится пересекать устройство 3-го уровня (маршрутизатор) для получения доступа к 80% ресурсов, она позволяет использовать в сети последовательный детерминированный способ получения доступа к ресурсам.

Географическими виртуальными сетями значительно легче управлять, чем виртуальными сетями, устройства которых находятся в географически различных областях. Преимущества использования виртуальных сетей заключаются в следующем:

1. Виртуальные сети предоставляют эффективный механизм для управления изменениями в топологии сети и значительно снижают затраты, связанные с реконфигурированием концентраторов и маршрутизаторов. Пользователи виртуальной сети могут использовать одно и то же сетевое адресное пространство (т.е. IP-подсеть), независимо от их физического расположения. Когда пользователи виртуальной сети перемещаются из одного места в другое, до тех пор, пока они остаются в одной и той же VLAN и подсоединены к одному и тому же порту коммутатора, их сетевые адреса не изменяются. Изменение расположения пользователя требует лишь таких простых действий, как включение штекера пользователя в соответствующий порт коммутатора и его конфигурирование.
2. Широковещательные потоки данных не передаются за пределы виртуальной сети. В свою очередь, смежные порты не получают широковещательных данных, которые генерируются другими виртуальными сетями. Такой тип конфигурации значительно уменьшает общий объём широковещательных данных, освобождает полосу пропускания для действительно полезных данных и снижает общий уровень уязвимости сети в отношении широковещательных штормов. Одной из проблем совместно используемых виртуальных сетей является относительная легкость проникновения в них. Подключаясь к работающему порту, несанкционированный пользователь получает доступ ко всем данным, передаваемым в сегменте. Чем больше группа пользователей, тем большие возможности создаются для потенциального несанкционированного доступа. Одним из экономически эффективных и легко административно реализуемых способов повышения уровня безопасности в сети является использование дополнительной функции коммутатора – идентификации источника. При несанкционированном подключении к работающему порту коммутатор определяет несуществующий MAC-адрес устройства и не пропускает его пакеты, кроме этого, некоторые коммутаторы формируют сообщение для администратора. Второй способ повышения безопасности заключается в том, что все неиспользуемые порты объединяются в одну неработающую виртуальную сеть.

Виртуальные сети могут строиться на основе трёх основных базовых моделей:

- сети, базирующиеся на портах (статические);
- сети на основе MAC-адресов (динамические);
- сети, основанные на протоколах VLAN.

8.1. Статические виртуальные сети

Принадлежность устройства к статической виртуальной сети на основе портов проиллюстрирована на рис. 8.2. Конкретной виртуальной сети назначается порт, который не зависит от пользователя или системы, подсоединённой к данному порту. Это означает, что все пользователи, подсоединённые к данному порту, будут членами одной и той же виртуальной сети. Назначение портов виртуальным сетям обычно осуществляет сетевой администратор. Конфигурация порта в этом случае является статической (неизменной), и переключение порта на другую виртуальную сеть не может быть выполнено автоматически, это может сделать только администратор, после реконфигурирования коммутатора.

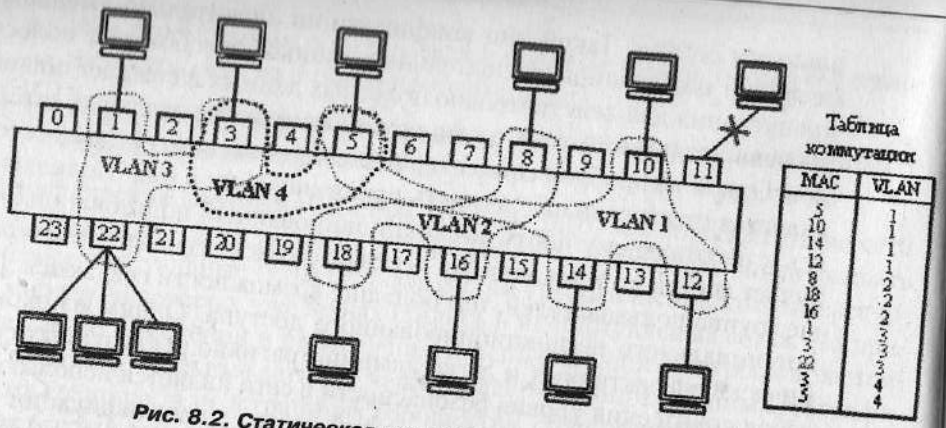


Рис. 8.2. Статическая виртуальная сеть на основе портов

Каждой виртуальной сети должен быть присвоен уникальный адрес, это помогает осуществлять коммутатору правильную доставку пакетов. Коммутатор содержит таблицу соответствия портов виртуальной сети, как показано на рис. 8.2. В приведённом примере организовано три виртуальных сети VLAN 1, VLAN 2 и VLAN 3. Поскольку порт 11 не принадлежит ни к одной из виртуальных сетей, то при подключении узла к порту 11 этому узлу в доступе к сети будет отказано. Соответственно, если, например, узел, подключенный к порту 16 (VLAN 2), пожелает передать информацию порту 10 (VLAN 1), то ему будет отказано в этой операции. Следует заметить, что не следует один порт включать в несколько виртуальных сетей. На рис. 8.2 приведён пример, когда администратор сети решил организовать ещё одну виртуальную сеть VLAN 4 всего для двух узлов 3 и 5. Но поскольку эти узлы принадлежат виртуальным сетям VLAN 3 и VLAN 1, то после организации сети VLAN 4 все узлы обеих виртуальных сетей будут находиться в рамках одной сети. То есть вместо трёх виртуальных сетей VLAN 1, VLAN 3 и VLAN 4 будет всего одна виртуальная сеть. Это достаточно типичная ошибка администраторов.

8.2. Динамические виртуальные сети

В динамической виртуальной сети автоматически определяется принадлежность узла к соответствующей виртуальной сети, с помощью таблицы MAC-адресов узлов, которая хранится в коммутаторе. Когда пользователь включает свой компьютер, коммутатор изучает MAC-адрес и проверяет таблицу на предмет наличия виртуальной сети, назначенной этому MAC-адресу. Этот процесс позволяет узлу оставаться в той же самой виртуальной сети независимо от того, к какому порту коммутатора он подключен.

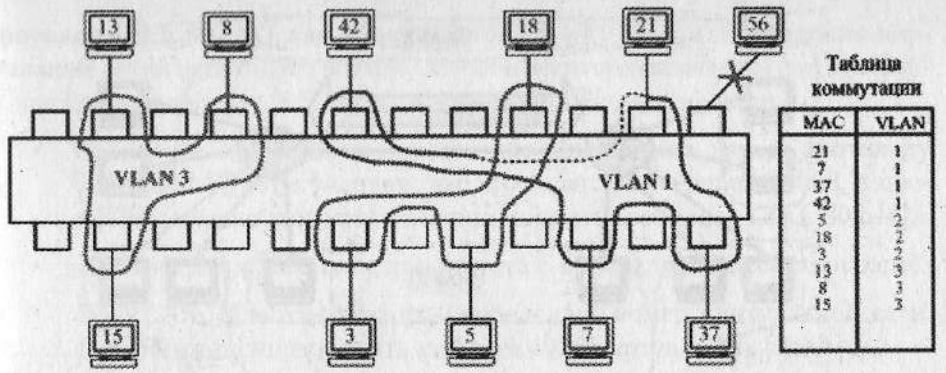


Рис. 8.3. Динамические виртуальные сети на основе MAC-адресов

На рис. 8.3 приведён пример трёх динамических виртуальных сетей. Поскольку узел с адресом 56 в таблице адресов коммутатора не зарегистрирован, то ему в доступе к сети будет отказано. Физическое перемещение узла не влияет на принадлежность к виртуальной сети, например, узел с адресом 13 может быть подключён к любому порту коммутатора, и при этом будет оставаться в той же виртуальной сети. Подключение нового узла к любой виртуальной сети требует только включения новой записи об этом узле и его принадлежности к соответствующей виртуальной сети в таблицу адресов коммутатора. Данный способ организации виртуальных сетей очень удобен в случае частого физического перемещения узлов, поскольку не требует дополнительных настроек коммутатора. Однако он совершенно не удобен в случае частой смены узлов, так как требует много усилий по управлению сетью, для поддержания таблицы физических адресов.

Можно использовать интеллектуальное программное обеспечение, чтобы сопоставлять номера виртуальных сетей аппаратным адресам, протоколам или даже адресам приложений, чтобы создавать динамические виртуальные сети.

8.3. Виртуальные сети, основанные на протоколах

При построении виртуальных сетей охватывающих несколько коммутаторов, как показано на рис. 8.4, коммутаторы должны каким-то образом различать циркулирующие по сети пакеты на предмет принадлежности той или иной виртуальной сети. В виртуальных сетях с несколькими коммутаторами

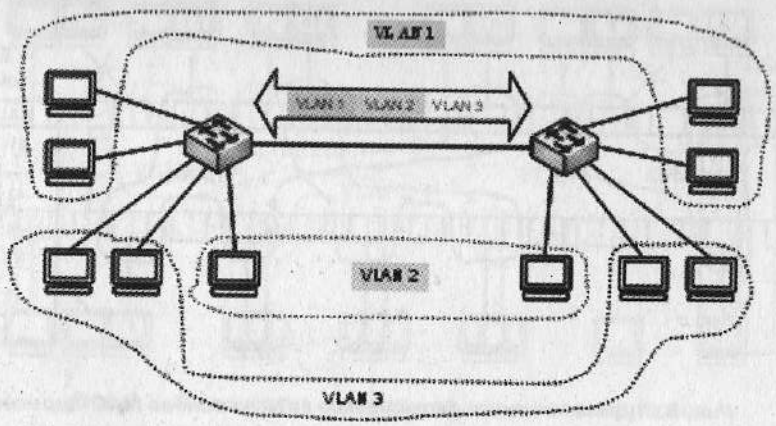


Рис. 8.4. Динамические виртуальные сети на нескольких коммутаторах

заголовки пакетов модифицируются для указания идентификатора ID (тега – признака) виртуальной сети до того, как они будет отправлены в канал между коммутаторами.

Перед отправкой пакета конечному устройству его заголовок изменяется и приобретает первоначальный вид. Идентификатор логически идентифицирует принадлежность пакета к определённой виртуальной сети. Для построения виртуальных сетей на нескольких коммутаторах используются четыре метода, или протокола, позволяющие отслеживать кадры виртуальных сетей:

- IEEE 802.10;
- IEEE 802.1Q;
- Inter-Switch Link (ISL);
- LAN Emulation (LANE).

Признаки пакетов в спецификации IEEE 802.10. Протокол 802.10 является стандартным методом идентификации виртуальных сетей путём вставки идентификатора в заголовок пакета. Этот процесс называется добавлением признака (тега). Протокол IEEE 802.10 используется для передачи информации виртуальным сетям, построенным на основе FDDI. Заголовок кадра FDDI содержит идентификатор безопасности (SAID), поле точки соединения (LSAP) и поле управления пространством (MDF). Поле SAID в заголовке кадра идентифицирует виртуальную сеть. В связи с тем, что сети FDDI в нашей стране встречаются крайне редко, то этот протокол детально рассматриваться не будет.

Протокол IEEE 802.1Q, как и протокол IEEE 802.10, для определения виртуальных сетей использует вставку идентификатора размером в четыре байта, как показано на рис. 8.5, который содержит следующие поля:

- Ether type. Это поле определяет принадлежность пакета протоколу IEEE 802.1Q, при условии, что его содержимое равно 81008, в противном случае этот пакет не принадлежит протоколу IEEE 802.1Q;
- PR. Трёхразрядное поле приоритета – определяет важность пакета;
- ID VLAN. Идентификатор – определяет номер виртуальной сети. Всего может существовать не более 4096 виртуальных сетей.

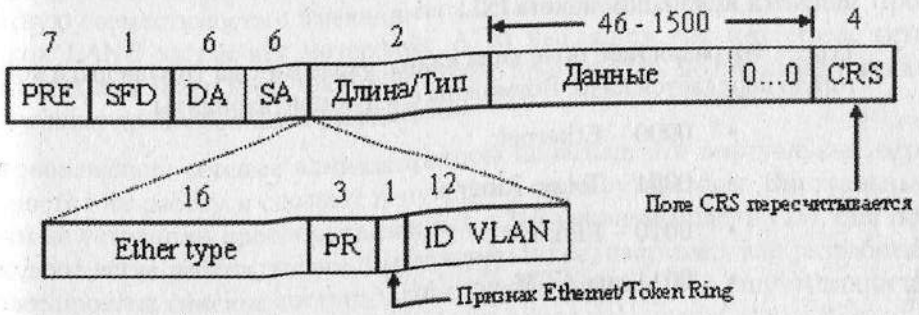


Рис. 8.5. Структура пакета данных протокола IEEE 802.1Q

Поскольку после вставки идентификатора в пакет его размер изменяется, то необходимо пересчитать контрольную информацию, а это в свою очередь требует достаточно больших вычислительных ресурсов, что может привести к дополнительным задержкам при передаче пакетов.

Протокол межкоммутационного канала Inter-Switch Link (ISL). Этот протокол является фирменным протоколом фирмы Cisco. Для идентификации пакетов, которые используется коммутаторами этой фирмы, используется обладающий низкой задержкой механизм мультиплексирования потоков данных от нескольких виртуальных сетей в один физический канал. Этот протокол был реализован для организации соединения между коммутаторами, маршрутизаторами и сетевыми адаптерами, используемыми на сетевых узлах, таких как серверы. Суть его заключается в том, что к стандартному кадру Ethernet добавляется заголовок размером 26 байт (208 бит), как показано на рис. 8.6.

При этом поле контрольной информации не пересчитывается. Заголовок LSI имеет следующие поля:

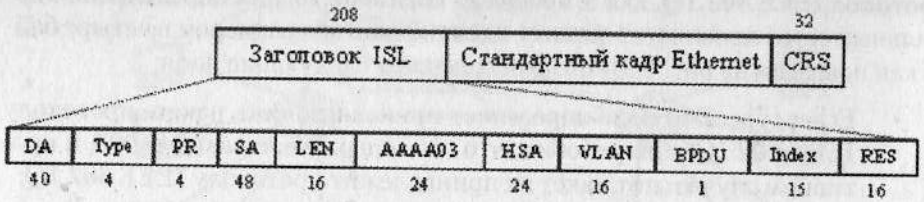


Рис. 8.6. Формат заголовка и кадра протокола ISL

- DA. Поле группового адреса получателя. Если 40 разрядов группового адреса установлены в 01-00-0C-00-00 или 03-00-0C-00-00, то он является признаком пакета ISL.
- Type – 4-х разрядное поле типа кадра, указывает на тип среды, в которую передаётся кадр. Возможны следующие варианты:
  - 0000 – Ethernet;
  - 0001 – Token Ring;
  - 0010 – FDDI;
  - 0011 для ATM.
- User – 4-х разрядное поле пользователя, используется для указания уровня приоритета кадра:
  - xx00 – нормальный приоритет;
  - xx01 – приоритет 1;
  - xx02 – приоритет 2;
  - xx11 – самый высокий приоритет.
- SA – 48-и разрядный адрес источника кадра.
- LEN – 16-ти разрядное поле. Это поле определяет длину пакета за исключением полей DA, Type, User, SA, LEN и CRC – всего 16 байт.
- AAAA03 – 24-разрядное стандартное поле для кадра LSI.
- HSA – старшая часть адреса источника кадра. Это поле в 24 разряда определяет изготовителя порта источника кадра.
- VLAN – 15-ти разрядный адрес виртуальной сети. Используется только 10 младших разрядов, что позволяет определять 1024 виртуальных сетей.
- BPDU – одноразрядная протокольная единица обмена мостов. Используется протоколами связующего дерева STP, а также для открытого протокола Cisco – CDP или танкового протокола виртуальных сетей VTP.
- Index – 16-ти разрядный индекс для указания адреса порта. Этот индекс может иметь любое значение и предназначен для использования в диагностических целях.
- RES – 16-ти разрядное резервное поле. Используется только в FDDI и Token Ring. В Ethernet оно заполнено нолями.

Протокол LAN Emulation (LANE). Этот протокол предназначен для транспортировки кадров виртуальных сетей по сетям ATM. Он обеспечивает обратную совместимость с имеющимися сетями Token Ring и Ethernet. Протокол LANE заставляет интерфейс ATM выглядеть как интерфейс сети Ethernet или Token Ring, так что не нужно производить модификации существующих драйверов или приложений.

Первоначально сетевые администраторы полагали, что виртуальные сети упростят их работу и сделают ненужными маршрутизаторы. Виртуальные сети не устранили проблем, связанных с 3-м уровнем модели OSI, они позволяют легче решать задачи 3-го уровня, такие, например, как разработка более простых списков доступа, однако необходимость в маршрутизации на 3-м уровне не исчезла.

8.4. Конфигурирование виртуальных сетей

Под статическими виртуальными сетями понимаются порты коммутатора, которым вручную назначаются виртуальные сети, путём использования управляющего программного обеспечения или непосредственным конфигурированием коммутатора. Эти порты поддерживают назначенную им конфигурацию виртуальных сетей до тех пор, пока она не будет изменена системным администратором. Хотя статические виртуальные сети требуют внесения изменений вручную, они безопасны, легко конфигурируются и удобны для мониторинга. Этот тип виртуальных сетей хорошо работает в сетях, в которых соблюдаются следующие условия:

- перемещения станций легко контролируются и управляются;
- имеется надёжное управляющее программное обеспечение для конфигурирования портов коммутатора;
- нежелательна дополнительная служебная нагрузка, требуемая для поддержки MAC-адресов конечных станций и создания новых таблиц фильтрации.

При конфигурировании статических виртуальных сетей на коммутаторах следует помнить следующие основные положения:

- максимальное количество подключаемых виртуальных сетей зависит от типа коммутатора и ограничивается количеством его портов;
- виртуальная сеть VLAN 1 является виртуальной сетью, создаваемой по умолчанию производителем;
- по виртуальной сети VLAN 1 рассылаются анонсирования маршрутов протокола обнаружения устройств и магистрального протокола;
- на всех коммутаторных магистралях, принимающих участие в работе виртуальных сетей, должен быть сконфигурирован один и тот же протокол инкапсуляции (поглощения);
- команды конфигурирования виртуальных сетей зависят от модели коммутатора;
- IP-адреса для некоторых моделей коммутаторов находятся в широковещательном домене виртуальной сети;
- при создании, добавлении и удалении виртуальных сетей коммутатор должен находиться в режиме VTP-сервера.

8.5. Магистральный протокол VTP

В радиотехнике под магистралью понимается отдельная линия связи, по которой передается информация нескольких радиоканалов.

В телекоммуникационных системах под магистралью понимается отдельный канал передачи между двумя точками, которыми обычно являются центры коммутации. Магистраль представляет собой физическое соединение, по которому проходят логические каналы, она является каналом типа «точка-точка», который поддерживает несколько виртуальных сетей.

Чем больше размер таблицы, которую требуется хранить коммутатору, тем медленнее становится процесс принятия решения об отправке кадров на соответствующие порты. Для эффективного управления передачей кадров от различных виртуальных сетей по одному физическому каналу или линии между двумя сетевыми устройствами требуется новый способ связи или язык коммуникации между этими двумя устройствами. Такой способ связи или протокол используется для того, чтобы эти устройства могли «договориться» о передаче и последующем распределении кадров на соответствующие порты на обоих концах магистрали. Для этой цели были созданы различные магистральные протоколы.

Протокол VTP (VLAN Trunking Protocol) был создан фирмой Cisco для того, чтобы распространять информацию и управлять конфигурацией виртуальной сети на всем протяжении коммутируемой сети. Например, для создания и поддержки соединений внутри виртуальных сетей, имеющих несколько связанных друг с другом коммутаторов, поддерживающих несколько виртуальных сетей, каждая из них должна быть сконфигурирована вручную на каждом коммутаторе. По мере роста организации и увеличения количества коммутаторов в сети, каждый новый коммутатор должен быть сконфигурирован вручную с вводом информации о виртуальных сетях. Всего лишь одно неправильное назначение в виртуальной сети может вызвать две потенциальных проблемы:

- перекрёстное соединение виртуальных сетей вследствие несогласованности в конфигурации виртуальных сетей;
- согласование и ликвидация противоречивости конфигураций в смешанной среде передачи, например в среде, включающей в себя сегменты Ethernet и FDDI.

Протокол VTP позволяет свести к минимуму возможную рассогласованность конфигураций, которая возникает при изменении топологии сети. Эта несогласованность может привести к снижению защищённости сети, поскольку перекрёстное соединение виртуальных сетей при использовании дублирующих имен может привнести к внутреннему разъединению при преобразовании одного типа виртуальной сети в другой (например, от Ethernet в FDDI). Использование протокола VTP предоставляет следующие преимущества:

- поддержка согласованности конфигураций виртуальных сетей во всей объединённой сети;
- поддержка схемы преобразования, которая позволяет виртуальной сети осуществлять магистральное соединение по смешанной среде, например преобразование виртуальной сети в магистраль ATM или FDDI;
- точное отслеживание и мониторинг виртуальных сетей;
- динамическое оповещение всех устройств сети о добавлении новой виртуальной сети;
- конфигурирование режима "Plug-and-play" при добавлении новой виртуальной сети.

Перед созданием на коммутаторе виртуальной сети необходимо сначала создать домен управления протокола VTP, в котором можно протестировать созданную виртуальную сеть. Все коммутаторы в одном и том же домене

26 байт	14 байт	3 байта	3 байта	Поле переменной длины		
Заголовок ISL	Заголовок Ethernet	Заголовок LLC	Заголовок SNAP	Заголовок VTP	Сообщение VTP	CRC

Рис. 8.7. Инкапсуляция сообщений протокола VTP в кадр протокола LSI

управления совместно используют информацию о виртуальных сетях. Коммутатор может присутствовать только в одном домене управления протокола VTP. Находящиеся в разных доменах коммутаторы не могут совместно использовать информацию протокола VTP. В протоколе VTP каждый коммутатор передаёт со своих магистральных портов следующую информацию:

- имя домена управления;
- номер версии конфигурации;
- известные виртуальные сети и их конкретные параметры.

Домен протокола VTP состоит из одного или более соединённых между собой устройств, совместно использующих доменное имя протокола VTP. Для передачи сообщений протокола VTP другим коммутаторам сети происходит инкапсуляция (преобразование путём поглощения) этих сообщений в кадры магистрального протокола, такого как ISL или IEEE802.1Q. На рис. 8.7 обобщенно показан процесс инкапсуляции данных протокола VTP в кадр протокола ISL.

Заголовок протокола VTP изменяется в зависимости от типа сообщения, однако в нём, как правило, присутствуют следующие четыре поля:

- версия протокола VTP – версия 1, 2 или 3;
- тип сообщения VTP – один из четырёх типов;
- длина имени домена управления – определяет длину имени домена, которое следует за этим полем;
- имя домена управления – в этом поле содержится имя домена управления, заданное в конфигурации.

Прокол VTP имеет три версии: версия 1, версия 2 и версия 3. Основных различий между двумя версиями немного, но они достаточно существенны, чтобы сделать их несовместимыми. Две версии не будут работать вместе в одной сети. Версия 1 по умолчанию поддерживает Ethernet. Версия 2 обеспечивает следующие дополнительные особенности, кроме поддержки Ethernet:

- проверку непротиворечивости;

- поддержку Token Ring;
- поддержку изменения прозрачного режима;
- поддержку нераспознанного поля TLV;
- если все коммутаторы в сети поддерживают VTP версию 2, то, если на одном из них установить версию 2, она распространится и на другие коммутаторы.

Версия 3 протокола VTP работает только на тех коммутаторах, на которых установлена операционная система фирмы Cisco. Помимо того, что эта версия поддерживает все функции предыдущих версий, она обеспечивает ещё следующие дополнительные функции:

- поддержку протяжённых виртуальных сетей;
- поддержку создания и объявлений частных виртуальных сетей;
- поддержку привязки виртуальных сетей и множественных покрывающих деревьев MST и распространение карты привязки;
- улучшенный сервер прав доступа;
- защиту от ошибок в базе данных, случайно возникающих в домене VTP;
- возможность конфигурирования через специальный порт.

Проверка непротиворечивости выполняется, если администратором введена новая информация через интерфейс CLI или через CNMP. Обычно проверка непротиворечивости не выполняется, если информация для каждого коммутатора получена через объявление VTP или считана из энергонезависимой памяти (NVRAM). Если информация получена через объявление или считана из NVRAM, то коммутатор проверит только дайджест объявления VTP; если есть противоречия, то будет сделана проверка непротиворечивости.

Поддержка Token Ring обеспечивается только в VTP версии 2 и 3. Она заключается поддержке коммутации сетей Token Ring и виртуальных сетей Token Ring.

Поддержка изменения прозрачного режима. В прозрачном режиме коммутаторы только переправляют сообщения и объявления и не добавляют никакую полученную новую информацию к собственной базе данных. Версия 1 позволяет коммутаторам проверять имя домена и версию перед пересылкой. Версии 2 и 3 позволяют коммутаторам переправлять сообщения и объявления VTP без проверки номера версии.

Нераспознанное поле TVL. Если получено объявление VTP, которое имеет неизвестное поле TLV, коммутаторы с протоколом VTP версии 2 и 3 будут распространять изменения через их транковые связи. VTP сервер или клиент распространяют изменения конфигурации по транковым связям, даже для тех VTP, которые невозможно интерпретировать. Неизвестное TLV сохраняется в памяти NVRAM.

8.5.1. Объявления протокола VTP

Коммутаторы в домене управления VTP обмениваются информацией о виртуальных сетях через объявления VTP. Существует три типа объявлений:

- **запрос объявление** – происходит, когда клиенты запрашивают информацию о виртуальной сети в текущей сети. VTP посылает объявление этого типа в ответ на запрос с соответствующими кратким и детальным объявлениями. Кадр объявления включает поле версии, поле кода, поле имени домена управления (до 32 байт) и поле начального значения;
- **краткое объявление** – посылается автоматически каждые 5 минут (300 секунд) всем коммутаторам сети. Краткое объявление может также быть послано, когда в сети происходит изменение топологии, например добавление или демонтаж коммутатора. Кадр краткого объявления содержит: поле версии, поле кода, поле последователей, поле имени домена управления, поле номера версии конфигурации, идентификатор обновления, отметку времени обновления и поле дайджеста;
- **детальное объявление** – содержит очень детальную информацию о сети, включая версию, код, номер последовательности, название домена управления, номер версии конфигурации и поля информации виртуальной сети.

VTP объявления могут содержать следующую информацию:

- **значения 802.105 SAID** – для физического носителя FDDI;
- **номер версии конфигурации** – чем больше номер, тем информация свежее;
- **названия эмулированных виртуальных сетей** – используются для ATM и LANE;
- **формат кадра** – информация о формате и содержании кадра;
- **название домена управления** – название домена управления VTP может быть длиной до 32 символов. Если коммутатор сконфигурирован для одного названия домена, а получает кадр с другим названием, то информация игнорируется;
- **дайджест** – применяется, когда в домене используется пароль, он может быть длиной от 8 до 64 символов. Пароль должен соответствовать паролю на получателе, иначе информация об изменениях игнорируется;
- **идентификатор обновления** – идентификатор коммутатора, который отправил краткое объявление;
- **конфигурация виртуальной сети** – включает известную информацию о виртуальной сети, специфические параметры и максимальный размер блока передачи для каждой виртуальной сети в домене управления VTP;
- **идентификатор виртуальной сети** – информация ISL или 802.1Q.

Объявление виртуальных сетей с заводскими установками основано на типах передающей среды. Порты пользователя не должны конфигурироваться как магистрали протокола VTP. Каждое объявление начинается как версия конфигурации с номером 0. По мере того, как в конфигурацию вносятся изменения, номер версии увеличивается на единицу + 1. Номер версии в домене управления продолжает возрастать до тех пор, пока он не станет равным 2^{31} = 2 147 483 648, после чего счетчик номера версии обнуляется.

Кадры-объявления посылаются по адресу групповой доставки, так что все VTP-устройства в одном и том же домене управления могут их получать. Кадры не пересылаются в соответствии с нормальными правилами обработки группового вещания коммутаторами. Все клиенты и серверы домена управления VTP модернизируют свои базы данных при каждом добавлении или удалении чего-либо в сети. Чтобы все члены домена управления VTP могли модернизировать свои базы данных, достаточно обновить только VTP-сервер при добавлении или удалении виртуальной сети.

Возможны два типа объявлений домена управления VTP:

- объявления, порожденные сервером;
- объявления-запросы от клиентов, нуждающихся в информации о виртуальной сети в связи с перезагрузкой или включением питания.

Каждое объявление имеет номер версии. Номер версии – одна из наиболее важных частей объявления VTP. Как только база данных VTP изменена, сервер VTP увеличивает номер версии на единицу (часто номер версии в технической литературе называют номером ревизии). Затем сервер VTP распространяет эту информацию из своей базы данных другим коммутаторам с обновлённым номером версии.

Когда коммутаторы получают объявление VTP, которое имеет более высокий номер версии, то они обновляют текущую информацию базы данных из NVRAM более свежей информацией. Если коммутатор получает объявление с более низким номером версии, чем у него, он думает, что имеет более свежую информацию, и игнорирует полученное объявление.

Когда новый номер версии рассылается по домену VTP, коммутаторы полагают, что самую новую информацию обо всех виртуальных сетях имеет самый высокий номер версии. Так, когда коммутаторы обнаруживают дополнительные виртуальные сети в объявлении VTP, они обрабатывают полученную информацию как подлинную.

Если новый коммутатор сконфигурирован как сервер и имеет номер версии выше, чем текущий номер версии, используемый в домене, то каждый член домена, получая информацию от него, должен переконфигурироваться. Такое событие может создать настоящее бедствие в сети, поскольку, если коммутатор видит более высокий номер версии, то он рассматривает эту информацию как более новую и заменяет существующую базу данных новой конфигурационной информацией, даже если это уничтожает информацию о виртуальной сети.

Многие администраторы сетей делают серьёзную ошибку, используя команду «уничтожения всей конфигурации», полагая, что она уничтожит текущий номер версии. Эта команда в действительности не «очищает всё». Прокол VTP имеет собственную память NVRAM, в которой сохраняется информация протокола, и в частности номер версии. Решить эту проблему можно двумя способами. Самый легкий путь состоит в выключении и включении питания коммутатора после перевода его в режим клиента. В режиме клиента происходит стирание информации в памяти NVRAM.

Другой путь состоит в том, чтобы сделать коммутатор клиентом и соединить его с сетью для того, чтобы получить новую версию, а затем конфигурировать его как сервер VTP.

Каждый раз, когда сервер рассылает обновлённое объявление, это увеличивает номер версии на единицу. Если коммутатор-клиент получает два объявления одновременно, то для использования он выбирает объявление с самым высоким номером версии.

По умолчанию управление доменами производится без использования мер безопасности; это означает, что коммутаторы обмениваются информацией без запроса паролей. Добавление пароля автоматически переводит домен управления в безопасный режим. Все коммутаторы домена должны иметь одинаковый пароль.

8.5.2. Режимы коммутаторов в протоколе VTP

Протокол VTP может быть сконфигурирован на коммутаторах в трёх режимах: режиме клиента, режиме сервера и прозрачном режиме.

Режим клиента позволяет коммутатору иметь те же самые функции, что и режим сервера, за исключением того, что он не может менять информацию о виртуальной сети. Коммутатор в режиме клиента не может создавать, изменять или удалять виртуальную сеть на любом клиенте протокола VTP или коммутаторе, кроме случая, когда он получает объявление от сервера. Однако он может объявлять свою собственную конфигурацию виртуальной сети, синхронизировать информацию о виртуальной сети с другими коммутаторами сети и участвовать в процессе отсечения VTP. Коммутаторы режима клиента получают информацию от сервера VTP в домене управления VTP. При выключении питания коммутатора вся глобальная информация о виртуальной сети теряется.

Режим сервера, который настроен по умолчанию, позволяет создавать, изменять и удалять виртуальные сети в домене управления. Изменения конфигурации рассылаются всем членам домена VTP. В домене управления VTP должен существовать, по крайней мере, один сервер VTP. Два или более коммутаторов могут быть настроены как серверы из соображений резервирования. При выключении питания коммутатор, сконфигурированный как сервер, сохраняет свою глобальную информацию о виртуальных сетях.

Прозрачный режим позволяет коммутаторам не принимать информацию VTP, а просто переправлять объявления, которые он получает, другим коммутаторам, участвующим в домене VTP. Несмотря на то, что коммутаторы, использующие прозрачный режим, будут посылать информацию и объявления VTP, они не будут модифицировать свои базы данных или рассылать информацию относительно изменений топологии или виртуальных сетей. В этом режиме коммутаторы не выполняют функции протокола VTP. VTP версии 2 и 3 позволяет коммутаторам, работающим в прозрачном режиме, переправлять объявления, которые они получает из любых транковых портов, позволяя другим коммутаторам, участвующим в VTP, получать объявления.

8.5.3. Отключение каналов в протоколе VTP

По умолчанию коммутатор распространяет по сети широковещательные данные, данные многоадресатной рассылки, неизвестные пакеты и одноадресные пакеты, рассылаемые методом лавинной рассылки, которые не являются необходимыми. Такое поведение приводит к дополнительной нагрузке на сеть.

Отключение протокола VTP увеличивает доступную полосу пропускания путём ограничения передачи данных на магистральные каналы, по которым должны передаваться данные соответствующим сетевым устройствам. По умолчанию функция отсечения протокола VTP выключена. Если на удалённом коммутаторе виртуальной сети нет доступного устройства, то отсечение протокола VTP предотвращает рассылку этим коммутатором данных этой виртуальной сети с магистрального порта, что предотвращает нерациональное использование полосы пропускания.

Когда отсечение VTP разрешено на сервере VTP, информация распространяется всем коммутаторам-клиентам и серверам в домене управления VTP. Этот шаг автоматически разрешает использование отсечения VTP и на этих коммутаторах. По умолчанию виртуальные сети с номерами от 2 до 1000 имеют право на отсечение VTP, а виртуальная сеть с номером 1 – никогда. Отсечение VTP обычно требует нескольких секунд, чтобы распространиться на других клиентов домена управления VTP после того, как этот режим был включен или питание коммутатора было выключено.

8.5.4. Настройка протокола VTP

Конфигурирование протокола сводится к выполнению следующих шагов:

1. **Конфигурирование версии протокола VTP**. В домене управления протокола VTP могут работать три версии протокола VTP: версии 1, 2 или 3. Первые две версии не могут взаимодействовать друг с другом. Если на одном из коммутаторов домена будет установлена версия 2 протокола VTP, то и все остальные коммутаторы этого домена управления должны быть сконфигурированы в этой версии или версии 3. По умолчанию конфигурируется версия 1 протокола. Установка 2 или 3-й версии может потребоваться в том случае, если есть необходимость в особых функциях, которые отсутствуют в версии 1.

Для того чтобы сконфигурировать версию протокола VTP на коммутаторе, использующем операционную систему IOS Cisco, требуется сначала войти в режим базы данных виртуальной сети.

2. **Конфигурирование домена протокола VTP**. Если устанавливаемый коммутатор является первым коммутатором в сети, то следует создать домен управления. Если же домен уже существует, то перед добавлением нового коммутатора следует проверить его имя. Если в домене управления введены меры безопасности, то в нём следует задать пароль. Имя домена может иметь длину от 1 до 32 символов и чувствительно к регистру. Пароль должен иметь длину от 8 до 64 символов.

При добавлении клиента протокола VTP к существующему VTP-домену обязательно следует проверить, что его номер версии конфигурации меньше, чем аналогичные номера у других коммутаторов данного VTP-домена. Коммутаторы VTP-домена используют конфигурацию виртуальной сети с наибольшим номером версии VTP-конфигурации. Если номер версии конфигурации на добавляемом коммутаторе окажется большим, чем номер версии в домене, то будет стерта информация о виртуальной сети сервера VTP и домена VTP.

3. **Конфигурирование режима протокола VTP**. На коммутаторе может быть установлен один из трёх доступных режимов протокола VTP.

Если устанавливаемый коммутатор первый в домене управления и будут добавляться другие коммутаторы, то на нём следует установить режим сервера. Другие добавляемые коммутаторы будут получать информацию о виртуальных сетях от этого коммутатора. В домене должен быть хотя бы один сервер.

Если коммутатор добавляется в домен управления, в котором уже имеются коммутаторы, то на этом коммутаторе следует установить режим клиента для предотвращения распространения им некорректной информации в уже существующей сети. Если предполагается в дальнейшем использовать этот коммутатор в качестве сервера, то изменить его режим на режим сервера следует после того, как он получит корректную информацию о виртуальных сетях от других устройств сети. Если для этого коммутатора не предполагается обмена информацией о виртуальных сетях с другими коммутаторами этой сети, то на нём следует установить прозрачный режим. В прозрачном режиме можно создавать, удалять и переименовывать виртуальные сети, не сообщая информацию об этих изменениях другим коммутаторам. Если конфигурирование устройств в сети выполняется несколькими пользователями, то существует риск наложения друг на друга виртуальных сетей с разными функциями, но с одними и теми же идентификационными данными.

Чтобы логически выделить виртуальную сеть или группу виртуальных сетей, которой принадлежат пакеты, коммутаторы Cisco помещают в кадры заголовки. Заголовки идентифицируют связанную с кадром виртуальную сеть; каждый коммутатор удаляет заголовок перед выходом кадра в порт связи доступа или в порты, принадлежащие идентифицированной виртуальной сети.

Коммутаторы не могут передавать кадры из одной виртуальной сети в другую. Для передачи пакетов между виртуальными сетями необходимо использовать внешний или внутренний процессор маршрутов. Внешний процессор маршрутов – это типичный маршрутизатор. Внутренний процессор

маршрутов представляет собой модуль внутри коммутатора, но коммутатор должен обращаться к нему как бы извне, чтобы осуществлять пересылку.

8.6. Протокол MSTP

Протокол Per VLAN Spanning Tree (PVST) создаёт уникальное связующее дерево для каждой виртуальной сети, что увеличивает количество вычислений, производимых коммутатором при наличии большого количества виртуальных сетей. Протокол множества связующих деревьев MSTP уменьшает эту нагрузку, позволяя использовать одно связующее дерево для нескольких виртуальных сетей.

Главная цель протокола MSTP состоит в том, чтобы уменьшить общее количество связующих деревьев, в соответствии с физической топологией сети, и таким образом уменьшить нагрузку центрального процессора коммутатора. Количество связующих деревьев уменьшается до числа доступных связей (активных путей).

В примере, показанном на рис. 8.8, протокол PVST+ мог бы создать до 1024 уникальных связующих деревьев, с собственными сообщениями BPDU, для обмена информацией с целью выбора корневого коммутатора и назначенных путей. Однако 1000 виртуальных сетей разделена на две группы по 500, каждая из которых использует только одну связь, соответствующую одному связующему дереву. В результате чего каждый коммутатор должен поддержать только два связующих дерева вместо 1000.

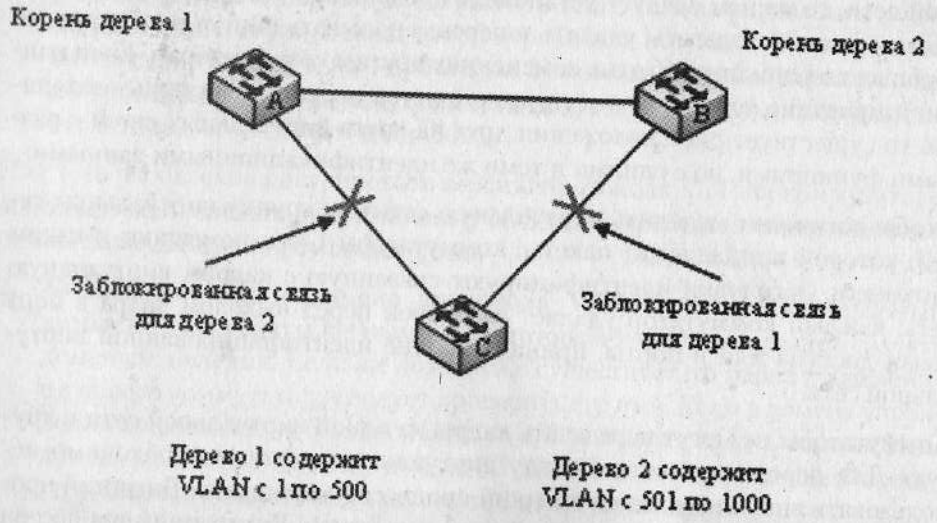


Рис. 8.8. Протокол связующих деревьев MSTP

Алгоритм протокола MSTP обладает более быстрой сходимостью по сравнению с протоколом PVST+ и совместим с протоколом STP; быстрым протоколом связующего дерева RSTP и протоколом Cisco PVST+.

Протокол MSTP позволяет строить множество связующих деревьев, группируя виртуальные сети по транковым каналам и связывая их со связующим деревом. В каждом конкретном случае используется топология виртуальной сети, не зависящая от топологии связующего дерева. Такая архитектура позволяет использовать несколько активных путей для передачи данных, что обеспечивает сбалансированность нагрузки линий связи.

Поскольку отказ связи в одной виртуальной сети не обязательно затрагивает другие виртуальные сети, то отказоустойчивость сети повышается. В больших сетях можно более легко управлять сетью, используя избыточные связи между виртуальными сетями.

Все коммутаторы должны быть сконфигурированы одинаково, что позволит им участвовать в определённом наборе связующих деревьев – зоне.

В примере, приведённом на рис. 8.9, протокол MSTP объединяет виртуальные сети в зоны, в которых виртуальные сети используют логические связующие деревья. Проблема соответствия виртуальных сетей связующим деревьям решается с помощью сообщений BPDU, в которых указывается идентификатор виртуальной сети и идентификатор связующего дерева.

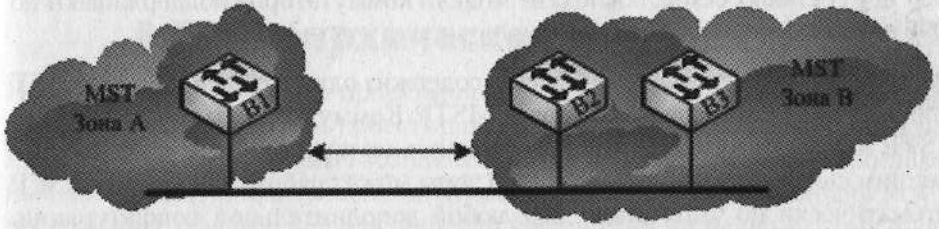


Рис. 8.9. Зоны MST

Чтобы обеспечить логическое соответствие виртуальной сети связующему дереву, каждый коммутатор, поддерживающий протокол MSTP, работает с оригинальной информацией, содержащей:

- алфавитно-цифровое название области (зоны) действия MSTP размером в 32 байта;
- номер ревизии зоны в 2 байта, определяющий время создания наиболее свежей информации;
- таблицу, содержащую до 1024 записей о виртуальных сетях, в которой указано, какому связующему дереву соответствует данная виртуальная сеть.

Для определения точных границ зоны необходимо передать не только название зоны и номер ревизии, но всю таблицу о виртуальных сетях. После того, как коммутатор получает сообщение BPDU, он извлекает номер сети и просматривает свою таблицу виртуальных сетей. Если номера сетей не совпадают, то отправивший сообщение коммутатор принадлежит другой зоне (коммутаторы расположены в разных зонах), а порт, на который было получено сообщение BPDU, расположен на границе зоны. На рис. 8.9 коммутатор B1 принадлежит зоне A, тогда как коммутаторы B2 и B3 находятся внутри зоны B.

Как и в протоколе PVST, 12-ти разрядное поле расширения в идентификаторе коммутатора используется и в протоколе MSTP, оно определяет номер виртуальной сети.

Каждый коммутатор должен иметь уникальный идентификатор. В PVST каждая виртуальная сеть связана с логическим портом коммутатора. Поэтому каждая виртуальная сеть должна иметь уникальный идентификатор коммутатора. MAC-адрес коммутатора не может быть использован для идентификации виртуальных сетей, поскольку не позволяет идентифицировать более одной виртуальной сети. Уникальность идентификатора виртуальной сети обеспечивается 16-разрядным полем приоритета, которое в свою очередь содержит 12-ти разрядный адрес (номер) виртуальной сети и 4-х разрядный приоритет. Коммутаторы фирмы Cisco сначала поддерживали до 1024 виртуальных сетей, последние модели коммутаторов поддерживают до 4096 виртуальных сетей.

Зона связующих деревьев MST всегда содержит одно внутреннее дерево IST и произвольное количество деревьев MSTP. Коммутатор, поддерживающий MSTP, должен иметь возможность общения, по крайней мере, с одним внутренним связующим деревом. Внутри зоны может работать протокол RSTP автоматически по умолчанию, без любой дополнительной конфигурации. Внутреннее связующее дерево взаимодействует с внешними по отношению к зоне устройствами. Внутри зоны связующие деревья MSTP обмениваются сообщениями BPDU, которые за зону не передаются. Взаимодействие с внешними устройствами осуществляется только с помощью внутреннего связующего дерева IST, которое работает на всех коммутаторах зоны. В результате чего вся зона, охватываемая внутренним связующим деревом, рассматривается как один коммутатор и управляется единственным внутренним связующим деревом CST. Внутреннее связующее дерево всегда имеет номер 0, а остальные нумеруются произвольно с 2 до 15.

На рис. 8.10 приведены две функционально эквивалентные диаграммы. В этой сети следует заблокировать порты между коммутаторами M и B и один из портов внутри сети. Однако из-за внутреннего связующего дерева IST вся

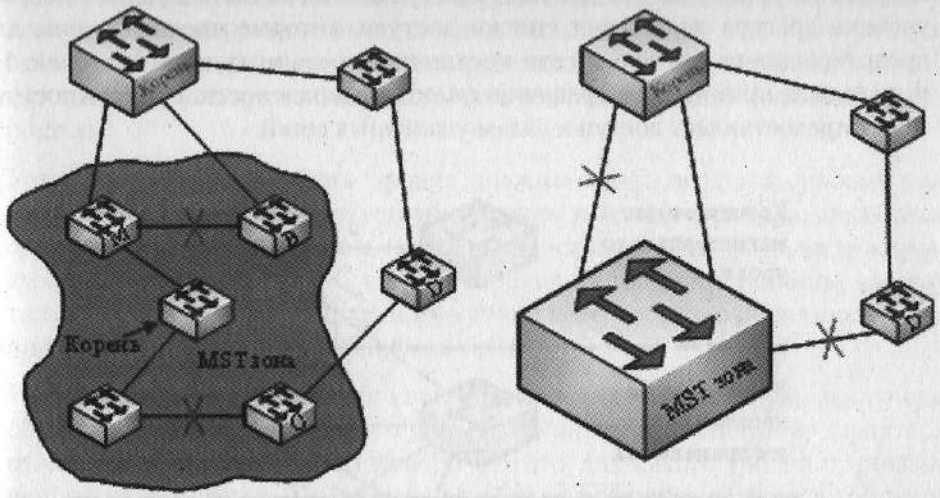


Рис. 8.10. Организация зоны MST

область MST рассматривается как один коммутатор. В результате чего следует заблокировать дополнительный порт на коммутаторе B и порт на коммутаторе D.

8.7. Иерархическая модель сети

Иерархический подход к проектированию сети состоит в разделении достаточно сложной задачи проектирования на ряд меньших и легче решаемых задач. Каждый уровень возникающей структуры решает свой собственный набор задач, что позволяет оптимизировать используемое аппаратное и программное обеспечение для решения конкретных, присущих этому уровню задач. Устройства самого нижнего уровня предназначены для приёма данных и передачи их в сеть для последующей обработки на более высоких уровнях. Для решения задачи проектирования сетей принят трёхуровневый подход. В этой трёхуровневой модели все сетевые устройства и соединения между ними группируются и подразделяются на три уровня: базовый (магистральный) уровень, уровень распределения и уровень доступа, как показано на рис. 8.11.

Уровень Доступа. На уровне доступа происходит передача данных в сеть и осуществляется входной контроль. Через этот уровень конечные пользователи получают доступ к сети. Коммутатор уровня доступа соединяет физический канал от интерфейса конечного пользователя до устройств, располо-

женных на уровне распределения. Выступая в качестве «парадной двери», уровень доступа использует списки доступа, которые предназначены для предотвращения доступа к сети несанкционированных пользователей. На этом уровне принимаются решения политик безопасности. Уровень доступа также предоставляет доступ к узлам удалённых сетей.

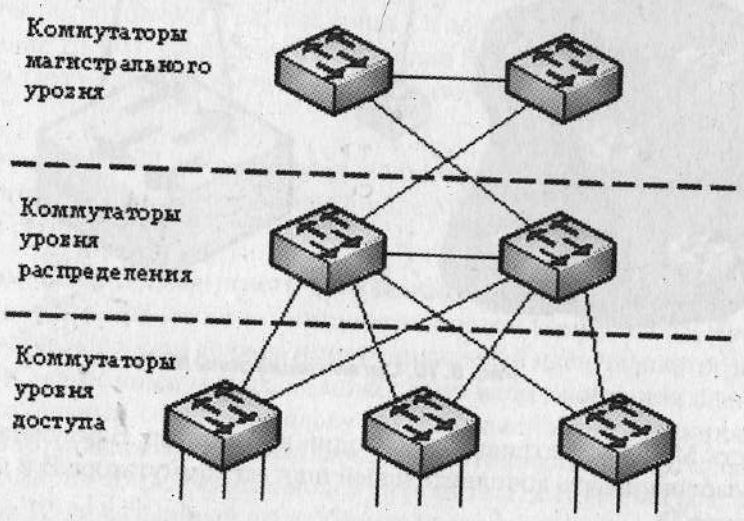


Рис. 8.11. Иерархическая модель сети

Уровень распределения определяет границы сети и обеспечивает манипуляцию пакетами в сети. Он расположен между уровнем доступа и магистральным уровнем. Его назначение состоит в отделении процессов магистрального уровня от остальной части сети. В частности, он должен создать границу входа в сеть путём использования списков доступа, определения широковещательных доменов, безопасности, управления размерами таблиц коммутации, обобщение (агрегацию) адресов сети, распределения статических маршрутов, перераспределения динамических маршрутов, соединений с удалёнными площадками и перераспределения потока информации между доменами. Таким образом, этот уровень определяет политику доступа к сети. Различные политики могут быть использованы для обеспечения безопасности сети и для экономии ресурсов путём предотвращения передачи нежелательных данных.

Магистральный уровень предназначен для создания оптимизированной и надёжной транспортной структуры для передачи данных с большими скоростями. Иными словами, магистральный уровень должен передавать данные максимально быстро, а само устройство должно быть очень надёжным

и содержать самые быстрые процессоры в сети. Устройства этого уровня не должны быть загружены выполнением таких операций, как проверка списков доступа, шифрование данных, трансляция адресов и других функций, которые препятствуют коммутации пакетов с максимально возможной скоростью.

Устройства магистрального уровня должны иметь доступ к любому узлу сети. Это не подразумевает, что они должны иметь физическую связь непосредственно с каждым узлом, но все устройства должны быть достижимы в таблице маршрутизации. С целью сокращения размеров таблицы маршрутизации устройств магистрального уровня целесообразно использовать технологию суммирования маршрутов сети.

На каждом уровне требуется свой тип коммутатора, который наилучшим образом решает задачи данного уровня. Функции и технические характеристики каждого коммутатора зависят от того, для какого уровня предназначен этот коммутатор. Правильный выбор наилучшего для данного уровня коммутатора обеспечивает максимальную эффективность работы сети для её пользователей. Это не означает, что обязательно следует использовать устройства на всех трёх уровнях.

Часто ошибочно полагают, что три рассматриваемых уровня (магистральный, распределения и доступа) должны быть явно выражены конкретными физическими устройствами, однако это не является обязательным. На самом деле эти три уровня определяются для успешного проектирования сети и чёткого определения функций, которые должна выполнять сеть. Конкретный способ реализации этих уровней зависит от потребностей проектируемой сети. Однако важно помнить, что для того, чтобы сеть оптимально функционировала и сохраняла масштабируемость по мере её роста, в процессе её проектирования должен быть соблюден иерархический принцип.

Иерархическая модель подразумевает избыточность, которая предотвращает воздействие отказа в одной части сети на всю сеть.

8.7.1. Коммутаторы уровня доступа

Уровень доступа выполняет следующие функции:

- предоставление совместно используемой полосы пропускания;
- предоставление коммутируемой полосы пропускания;
- фильтрация на MAC-уровне;
- микросегментация.

Как показано на рис. 8.11, коммутаторы уровня доступа функционируют

на 2-м уровне эталонной модели OSI и, в частности, предоставляют службу виртуальных локальных сетей (VLAN). Главной целью коммутатора уровня доступа является соединение конечных пользователей с сетью. Коммутатор уровня доступа должен выполнять эту функцию с минимальными затратами и максимальной плотностью портов.

Если две или более рабочих станций или два сервера непосредственно подсоединены к портам коммутатора, то вся полоса пропускания соединения с коммутатором предоставляется компьютерам, осуществляющим соединение. Если два или более компьютера подсоединены к концентратору, то каждому компьютеру предоставляется полоса пропускания, равная общей полосе пропускания, поделённой на количество подсоединённых к концентратору компьютеров. Концентратор может быть подсоединён к порту коммутатора. В этом случае полоса пропускания используется совместно всеми устройствами, подсоединёнными к порту коммутатора через концентратор.

Трёхуровневую модель впервые предложила фирма Cisco. В настоящее время такой метод построения сложных сетей стал стандартом "де факто". Для уровня доступа выпускается огромное количество коммутаторов различными компаниями. Например, отечественная компания ELTEX выпускает стековые коммутаторы уровня доступа MES1024 и MES2124.

Модель MES1024 является эффективным устройством доступа в небольших или средних сетях. Использование модели MES2124 целесообразно для обеспечения доступа серверов и пользователей, которым требуется широкая полоса пропускания. Это становится возможным благодаря наличию у этой модели нескольких портов Gigabit Ethernet.

8.7.2. Коммутаторы уровня распределения

Уровень распределения расположен между уровнем доступа и магистральным уровнем и предназначен для определения и отделения базового уровня. Его функция состоит в задании границ, в которых может происходить какая-либо обработка пакетов. На этом уровне происходит сегментация сети на отдельные широковещательные домены. Здесь же с помощью списков доступа может определяться политика и производиться фильтрация пакетов. Этот уровень обеспечивает изоляцию основной части сети от проблем, которые могут возникать в рабочих группах, с тем чтобы они не затрагивали магистральный уровень. Коммутаторы этого уровня могут работать на 2-м и 3-м уровнях эталонной модели OSI. По этой причине коммутаторы этого уровня часто называют многоуровневыми коммутаторами. Они соединяют в одном устройстве функции маршрутизатора и коммутатора, однако предназначены для коммутации потоков данных с большей скоростью, чем это

делает обычный маршрутизатор. Если в таких многоуровневых коммутаторах отсутствует встроенный маршрутизирующий модуль, то для выполнения функций 3-го уровня используется внешний маршрутизатор.

Устройствам уровня распределения требуется меньшее количество интерфейсов и меньшая скорость коммутации, чем коммутаторам магистрального уровня, поскольку им приходится обрабатывать меньшие объёмы данных. Тем не менее, молниеносная скорость на магистральном уровне становится бесполезной, если низкая скорость уровня распределения не позволяет пользователям получать быстрый доступ к нему.

Уровень распределения отделяет друг от друга уровень доступа и магистральный уровень и помогает четко очертить их соответствующие функции. В среде кампуса уровень распределения может выполнять несколько функций в частности:

- агрегирование адресов или зон;
- управление доступом на уровне отдела или рабочей группы;
- определение границ широковещательной и многоадресной рассылки;
- управление маршрутизацией сетей VLAN;
- все переходы из одной среды передачи в другую;
- обеспечение безопасности.

В средах, не являющихся сетями кампусов, уровень распределения может быть точкой перераспределения между доменами маршрутизации или точкой демаркации между протоколами статической и динамической маршрутизации. Он может также быть точкой доступа с удалённых узлов к корпоративной сети.

В целом уровень распределения можно охарактеризовать как уровень, на котором обеспечиваются соединения пользователей в соответствии с принятой в сети политикой.

Для работы на уровне распределения компания ELTEX выпускает следующие два семейства коммутаторов уровня L3: MES3100 и MES3100F. Все коммутаторы имеют 8, 16 или 24 Гигабитных порта и от 2-х до 4-х 10 Гигабитных портов.

8.7.3. Коммутаторы магистрального уровня

Поскольку магистральный уровень является центральной частью сети, он должен быть спроектирован для быстрой и надёжной работы. Использование на

магистральном уровне списков доступа нежелательно, поскольку они вносят дополнительную задержку. Более того, сам доступ конечных пользователей к магистральному уровню весьма нежелателен. Коммутаторы этого уровня могут использовать ряд технологий 2-го уровня. В иерархической сети данные конечных пользователей должны попадать на коммутаторы магистрального уровня только после того, как они прошли уровни доступа и распределения, где к ним могут быть применены списки доступа. При проектировании сети в качестве магистрального уровня может быть использован уровень маршрутизации (3-й уровень). Коммутаторы этого уровня должны при необходимости обеспечивать эффективное выполнение функций 3-го уровня. Поскольку маршрутизация на магистральном уровне осуществляется без использования списков доступа, трансляции адресов и других операций с пакетами, может показаться, что для выполнения столь простой задачи достаточно даже и маломощных коммутаторов. Однако верно как раз обратное – на этом уровне используются самые мощные коммутаторы/маршрутизаторы, поскольку в них используются самые быстрые технологии коммутации.

Одним из способов обеспечения надёжности, используемым в устройствах магистрального уровня, является использование избыточных каналов. При наличии возможности эти избыточные каналы следует делать симметричными (т.е. с одинаковой пропускной способностью), с тем чтобы можно было использовать распределение (балансирование) нагрузки.

Другим способом обеспечения надёжности является использование избыточных источников питания. Магистральные устройства обычно имеют два или более «заменяемых на ходу» источника питания, которые могут быть удалены или заменены индивидуально без выключения питания устройства.

Магистральные каналы оказываются самыми быстрыми, самыми надёжными, но и самыми дорогостоящими выделенными соединениями в распределённых сетях. Использование мощных устройств и каналов может сделать магистральный уровень сети крайне дорогостоящим. Для уменьшения расходов некоторые проектировщики сетей выбирают вариант несимметричных каналов. Вместо избыточных каналов в качестве резервных могут быть использованы технологии коммутации пакетов или коммутации по требованию. Компромиссное решение, позволяющее сэкономить средства при использовании таких технологий, определяется достигаемым уровнем производительности сети.

В некоторых случаях магистраль локальной сети также может быть причислена к магистральному уровню. В сетях кампусов или в крупных сетях, охватывающих комплекс офисов или смежных зданий, магистраль может проходить по локальной сети. В этом случае в качестве технологий магистрального

уровня типичным является использование Gigabit Ethernet и 10Gigabit Ethernet, а иногда и 40Gigabit Ethernet на основе оптоволоконного кабеля. Такие коммутаторы принимают на себя основную нагрузку в магистралях локальных сетей, поскольку они коммутируют пакеты на 2-м уровне со значительно большей скоростью, чем маршрутизаторы коммутируют пакеты на 3-м уровне.

Как было описано выше на протяжении нескольких глав, коммутаторы наделяются большим количеством дополнительных функций. Условно их деление можно представить на рис. 8.12.

Коммутаторы, осуществляющие идентификацию источника, подавление источника и поддерживающие алгоритм связующего дерева, относятся к коммутаторам первого уровня.

Коммутаторы, реализующие все функции коммутаторов первого уровня и кроме этого осуществляющие поддержку виртуальных сетей, относятся к коммутаторам второго уровня.

А коммутаторы, реализующие все функции, представленные на рис. 8.12, относятся к коммутаторам третьего уровня. Это самые сложные и дорогостоящие устройства.

Помимо разного количества функций, которые выполняют коммутаторы, они могут содержать разное количество портов различной производительности и быть выполнены в различных конструктивных вариантах, что даёт огромное количество вариантов их исполнения, многократно отличающихся в цене. Например, простейший коммутатор, не поддерживающий никаких дополнительных функций и имеющий 16 портов со скоростью передачи информации в 10 Мбит/с, может иметь стоимость около $200.

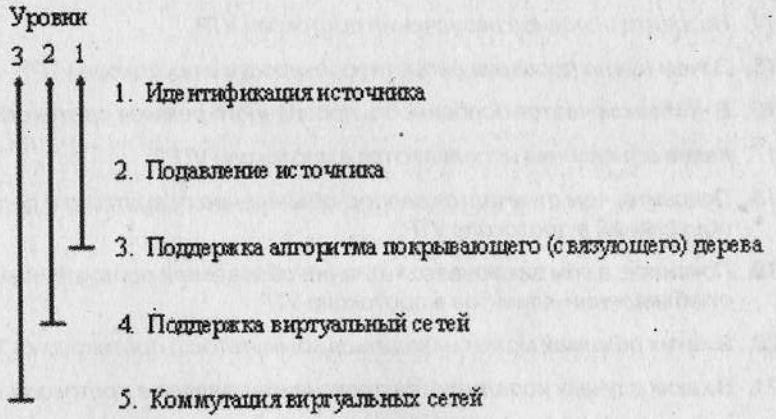


Рис. 8.12. Деление коммутаторов на уровни

А модульный коммутатор третьего уровня, имеющий 1024 порта, у которого все порты работают со скоростью Gigabit, а несколько – со скоростью 10 Gigabit и 40 Gigabit, может иметь стоимость более $100 000.

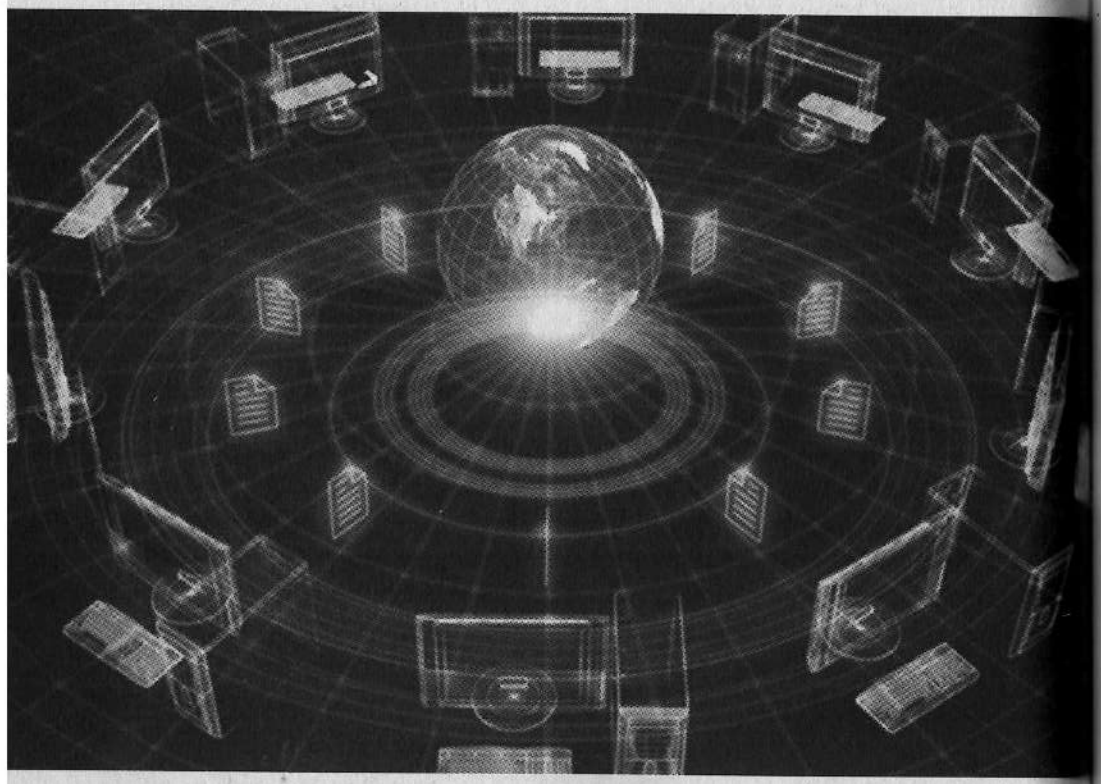
Контрольные вопросы

1. *Зачем нужна сегментация в сети?*
2. *В чём заключается отличие традиционной сегментации от сегментации с помощью виртуальных сетей?*
3. *Что такое сквозные виртуальные сети?*
4. *Что такое географические виртуальные сети?*
5. *Каким образом строятся статические виртуальные сети?*
6. *В чём заключается отличие динамических виртуальных сетей от статических?*
7. *Каким образом коммутаторы определяют, что передаваемый пакет принадлежит виртуальной сети?*
8. *Какие признаки используются в спецификации IEEE 802.10?*
9. *Каким образом в спецификации IEEE 802.1Q производится идентификация виртуальной сети?*
10. *В чём заключается особенность протокола ISL?*
11. *Назовите особенности протокола LANE.*
12. *В чём заключается конфигурирование виртуальных сетей?*
13. *Поясните, зачем нужны магистральные протоколы.*
14. *Назовите основное назначение протокола VTP.*
15. *Зачем нужна проверка непротиворечивости в протоколе VTP?*
16. *В чём заключается особенность прозрачного режима протокола VTP?*
17. *Какие объявления используются в протоколе VTP?*
18. *Поясните, чем отличается запрос-объявление от краткого и дельного объявлений в протоколе VTP.*
19. *Поясните, в чём заключается отличие объявлений порождённых сервером от объявлений-клиентов в протоколе VTP.*
20. *В каких режимах может находиться коммутатор с протоколом VTP?*
21. *В каких случаях используется отключение каналов в протоколе VTP?*
22. *В чём заключается конфигурирование версии протокола VTP?*

23. *Чем вызвана необходимость конфигурирования домена в протоколе VTP?*
24. *В чём заключается конфигурирование режима протокола VTP?*

Глава 9.

ЧАСТНЫЕ ВИРТУАЛЬНЫЕ СЕТИ



Многие компании и организации имеют удалённые филиалы и офисы, для связи с ними можно использовать индивидуальные каналы связи, однако при большой протяжённости линий это обходится крайне дорого. Для связи с удалёнными объектами можно использовать открытые каналы связи, но для сохранения конфиденциальности информации необходимо применять сложные и дорогостоящие методы шифрования информации. В настоящее время появилась возможность организовать частную сеть на основе сетей массового доступа, такие сети получили название - виртуальная частая сеть VPN. Она представляет собой сеть, создаваемую в открытой сетевой инфраструктуре, такой, например, как глобальная сеть Internet.

Преимущества использования частных виртуальных сетей по сравнению с использовавшимися ранее методами канального шифрования и другими альтернативными решениями следующие:

1. Одна технология VPN может обеспечить конфиденциальность нескольких приложений протокола TCP/IP. Шифрование на уровне приложений требует использования различных методов для разных служб. Возможна также поддержка нескольких протоколов посредством IP-туннелирования с использованием протоколов туннелирования типа "точка-точка", L2TP или протокола пересылки на 2-м уровне. Обеспечение конфиденциальности для нескольких приложений протокола TCP/IP особенно важно в ситуациях, когда необходимо обеспечить безопасный доступ для партнеров или удалённых работников.
2. Службы шифрования могут быть предоставлены для всех соединений TCP/IP между надёжным клиентом и сервером частной VPN сети. Преимуществом такого подхода является его прозрачность для конечного пользователя. Поскольку шифрование включено, сервер может повысить уровень шифрования.

9.1. Типы частных виртуальных сетей

Виртуальная частная сеть VPN (рис. 9.1) представляет собой множество соединений пользователей, установленных в совместно используемой инфраструктуре, с теми же политиками, как и в частной (локальной) сети. Эта совместно используемая инфраструктура может включать в себя уже имеющуюся магистраль провайдера службы, например, Internet. На практике используются три типа VPN-сетей:

VPN-сети удалённого доступа. Эти сети предоставляют удалённый доступ пользователям корпоративной интранет- или экстранет-сети с использованием тех же политик, какие используются в частой сети. VPN-сети удалённого доступа, при необходимости, предоставляют пользователям доступ к корпоративным ресурсам в любое время, в любом месте и любым способом. Для обеспечения безопасного соединения мобильных пользователей и филиалов предприятия VPN сети удалённого доступа могут использовать аналоговую телефонную линию, сеть интегрированных цифровых служб ISDN, цифровые абонентские каналы DSL, мобильные IP-телефоны и кабельные технологии.

Частные виртуальные интранет-сети. Такие сети соединяют между собой головной офис компании, удалённые офисы и филиалы через совместно используемую инфраструктуру с использованием выделенных линий. В этом случае пользователи получают все используемые в частной сети политики, включая обеспечение безопасности, качество обслуживания QoS, управляемость и надёжность.

VPN экстранет-сети. Эти сети соединяют между собой имеющих общие интересы пользователей, партнеров или сообщества в корпоративную интранет-сеть через совместно используемую инфраструктуру с использо-

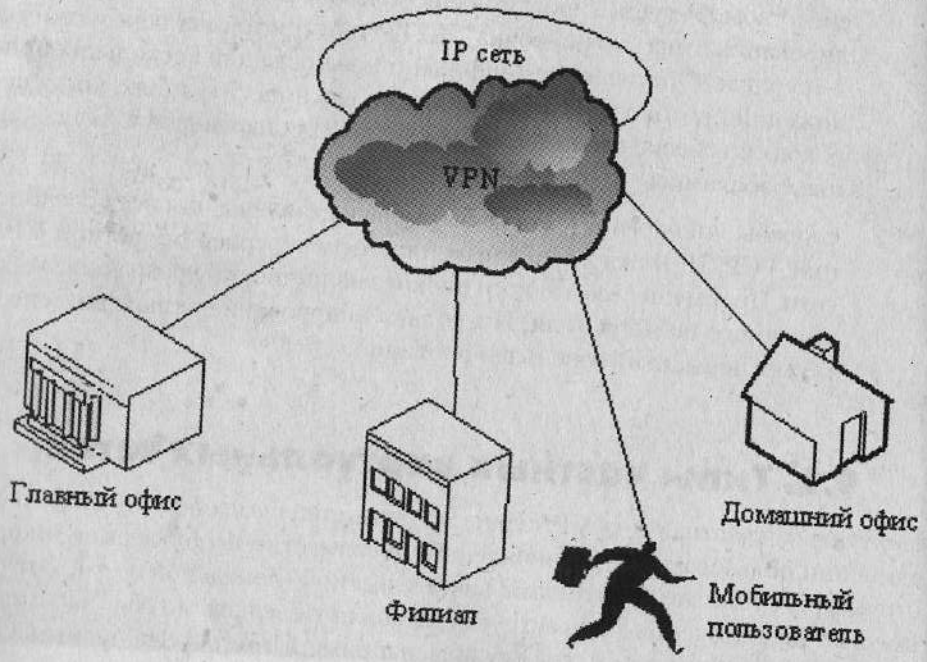


Рис. 9.1. Логическая топология VPN-сети

ванием выделенных линий. В этом случае пользователи получают все используемые в частной сети политики, включая обеспечение безопасности, качество обслуживания, управляемость и надёжность.

Соглашения об уровне обслуживания представляют собой договор между провайдером VPN-сети и его клиентом; в этом соглашении содержатся критерии, согласно которым пользователь ожидает предоставления конкретных служб. Соглашение об уровне обслуживания является единственным имеющим юридическую силу средством, имеющимся в распоряжении пользователя, которое позволяет ему добиться того, чтобы провайдер VPN-сети предоставил ему службы того уровня и качества, которые определены в соглашении. Однако в этих соглашениях имеются некоторые аспекты, представляющие определённые технические проблемы, как для провайдера, так и для его клиента.

Для клиента определённую проблему составляет выбор и установка технических средств, позволяющих оценить уровень обслуживания и, соответственно, степень соответствия предоставляемых услуг критериям соглашения. Следует отметить, что в одном соглашении могут содержаться положения, описывающие совместные контрактные обязательства нескольких провайдеров, но если VPN-сеть пользователя охватывает несколько доменов различных провайдеров, то в соглашении должен быть оговорён порядок взаимного соединения провайдеров и характеристики сквозного соединения.

Для провайдера проблемы, связанные с соглашением об уровне обслуживания, состоят в том, что их выполнение может быть связано с использованием услуг нескольких других провайдеров. В случае взаимодействия с Internet-провайдером в открытой сети типичный режим "негарантированной доставки" мало подходит для выполнения условий соглашения, в силу непредсказуемого характера работы механизмов выделения ресурсов. В таких средах провайдер должен либо обеспечить правильно спроектированную сеть (в смысле нужного соотношения между скоростью доступа пользователя и возможностями внутренней коммутации), либо создать структуры дифференциации служб для обеспечения того, чтобы каждому клиенту соглашений был предоставлен некоторый минимальный уровень выделения ресурсов. Следует отметить, что первый способ имеет тенденцию к уменьшению возможностей агрегирования потоков данных, что негативно влияет на конечную стоимость: второй способ оказывает неблагоприятное воздействие на сложность оперативного управления и на возможности масштабирования сети.

В примере, показанном на рис. 9.2, узлы сети А образуют VPN-сеть (показанную пунктиром) в магистральной сети провайдера службы, в то время

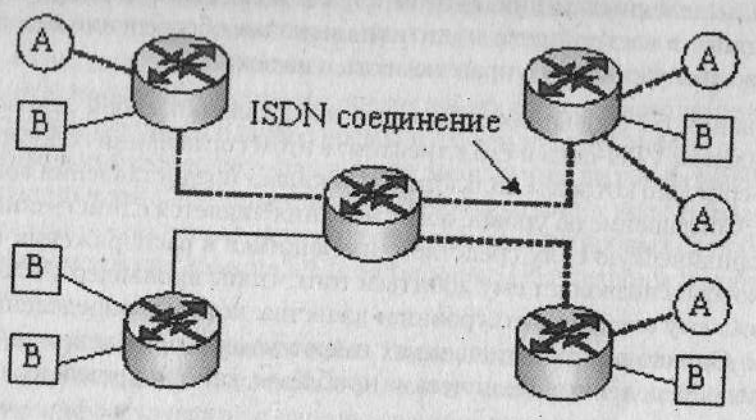


Рис. 9.2. Виртуальная сеть из узлов А

как сеть В даже не подозревает о её существовании. При этом сети А и В гармонично сосуществуют в одной и той же общей магистральной инфраструктуре.

Наиболее общим типом частной виртуальной сети является объединение географически удалённых друг от друга подсетей, принадлежащих общему административному домену, которые соединены между собой через совместно используемую инфраструктуру, находящуюся вне их административного контроля (например, через глобальную сеть Internet или магистральную сеть одного провайдера). Соединения между устройствами такой виртуальной VPN-сети могут оказаться легко уязвимыми, поэтому уровень конфиденциальности, обеспечиваемый частной виртуальной сетью, в значительной степени зависит от технологии, которая использовалась при её создании. Например, если данные, передаваемые между всеми подсетями (или всеми узлами VPN-сети), надёжно шифруются при прохождении по общей сетевой инфраструктуре, то обеспечивается относительно высокий уровень безопасности всей частной виртуальной сети.

Частные виртуальные сети позволяют отделить друг от друга различные типы потоков данных, с тем чтобы пользователи не знали о существовании друг друга. Виртуальные частные сети настолько независимы, что крупные поломки или неустойчивость в одном сообществе, абсолютно не влияют на другое сообщество.

Виртуальные частные сети могут охватывать более чем одну сеть с тем, чтобы её состояние поддерживалось одним или более провайдером, что обеспечивает повышенный уровень надёжности, поскольку все провайдеры явным образом поддерживают распределённую среду.

9.2. Реализация частных виртуальных сетей

После того, как принято решение о создании VPN-сети, необходимо принять соответствующие меры для обеспечения адекватной защиты инфраструктуры; для этого перед началом реализации необходимо провести аудит безопасности предлагаемых решений, строго определить приложение и цели его применения.

Аудит безопасности. Аудит безопасности системы проводится для того, чтобы после реализации VPN-сети уязвимость в брандмауэре, приложениях удалённого доступа или методе аутентификации не стала открытой более широкой аудитории, чем до реализации этой сети. Эти меры обеспечения безопасности заслуживают всестороннего изучения, по крайней мере, периметра сети, который может быть атакован как изнутри сети, так и извне. Такая оценка должна включать в себя аудит всех входных соединений в интранет-сеть, включая серверы удалённого доступа, соединения на базе маршрутизаторов и соединения с экстранет. В случае экстранет виртуальной частной сети такая оценка также должна включать анализ периметра сетей организаций-партнеров.

Сфера действия сети и требования приложений. После того, как выполнен аудит безопасности, необходимо определить сферу действия виртуальной частной сети путём идентификации конечных пользователей (таких как партнеры, подрядчики и мобильные абоненты), которые бы могли получать доступ к VPN-сети и приложениям.

Этот этап включает в себя оценку приложений, которые будут использоваться, и степень чувствительности данных к задержкам, похищению или искажению, которые они будут передавать. Оценка чувствительности приложений позволит выбрать соответствующие службы безопасности и адекватное шифрование.

Далее необходимо определить и протестировать задержку и потребности в качестве обслуживания QoS всех приложений, которые будут поддерживаться частной виртуальной сетью. Даже приложения, которые можно рассматривать как допускающие временное хранение в буфере с последующей отправкой, могут оказаться чувствительными к задержке. Эти уровни чувствительности влияют на то, в какой степени приложение является подходящим для частной виртуальной сети и поможет определить характер соответствующей сети.

Документация. В случае организации экстранет частной виртуальной сети, администрация предприятия должна зафиксировать правила поведения

пользователей в специальном документе. При использовании удалённого доступа этот документ должен отражать политику администрации по отношению к новой форме доступа пользователей.

Политика безопасности. В случае экстранет сети VPN, работа в виртуальной частной сети определяется политикой аутентификации и авторизации. Авторизация должна основываться на общих принципах, которые определяются работающими в данной сети предприятиями. Эти общие принципы должны иметь базовые привилегии – включая доступ к сети и доступ к Web-серверу – к уровням хранения продуктов поставщиков. Эти принципы должны быть позднее преобразованы в определённый набор данных и распространены по VPN-сети. Политика удалённого доступа в виртуальной частной сети не должна радикально отличаться от существующей. Наиболее важным отличием от традиционной политики удалённого доступа может быть "допустимое (приемлемое) использование" для доступа в Internet там, где это необходимо.

9.3. Проектирование VPN-сетей

Использование на всём протяжении маршрута прохождения данных качественного сетевого оборудования и программного обеспечения предоставляет возможности обеспечения:

- комплексной безопасности при передаче конфиденциальных данных по открытой инфраструктуре;
- требуемого качества обслуживания QoS, посредством дифференциации потоков данных;
- надёжности для критически важных приложений;
- масштабируемости для поддержки большой полосы пропускания данных;
- всеобъемлющего управления сетью для полного решения задачи доступа к виртуальной частной сети.

Ключевым компонентом службы виртуального удалённого доступа является туннелирование, которое представляет собой способ поглощения (инкапсуляции) пакетов протоколом, который понятен только во входной и выходной точках данной сети. Эти входная и выходная точки определяются как туннельные интерфейсы. Туннельный интерфейс аналогичен аппаратному интерфейсу, однако конфигурируется программно.

Используя туннелирование оригинального протокола фирмы Cisco L2TP,

Internet-провайдер или другая служба доступа могут создать виртуальный туннель для связи удалённых узлов пользователя с домашней корпоративной сетью. Концентратор доступа протокола L2TP, расположенный в точке присутствия провайдера, обменивается сообщениями протокола PPP с удалёнными пользователями. Посредством запросов и ответов протокола L2TP концентратор доступа LAC осуществляет связь с сетевым сервером LNS протокола L2TP для организации туннелей.

Протокол L2TP передаёт пакеты протокольного уровня по виртуальному туннелю между конечными точками соединения типа "точка-точка". Пакеты от удалённых пользователей принимаются точкой присутствия POP Internet-провайдера, преобразовываются в пакеты протокола L2TP и пересылаются по соответствующему туннелю.

Домашний шлюз пользователя принимает эти пакеты протокола L2TP, извлекает из него данные и пересылает их на соответствующий интерфейс. На рис. 9.3 подробно показаны отдельные элементы туннеля протокола L2TP и процесс подсоединения пользователя к серверу LNS для получения доступа к требуемой корпоративной Intranet-сети.

Как показано на рис. 9.3, при туннелировании используются три типа протоколов:

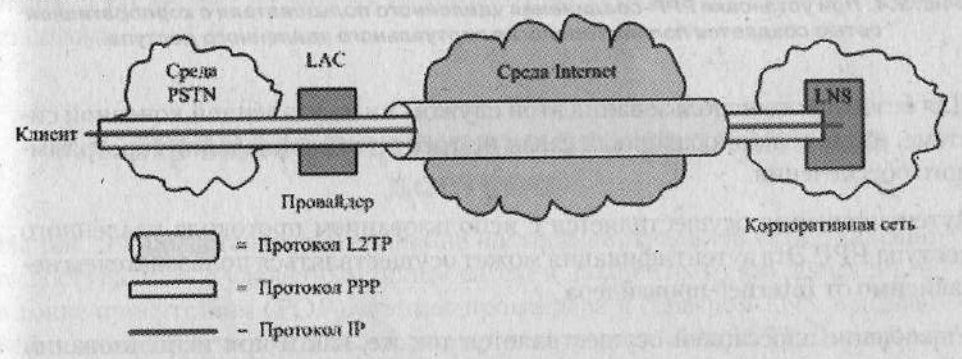


Рис. 9.3. Структура туннелирования протокола L2TP

Протокол пассажира, данные которого преобразуются в пакеты другого протокола, в качестве которого могут выступать протоколы PPP, SLIP и текстовый диалог.

Протокол преобразования (инкапсуляции) создаёт, поддерживает и ликвидирует туннель.

Протокол носителя осуществляет передачу по сети данных протокола преобразования (инкапсуляции). Чаще всего в качестве протокола носителя используется протокол IP, поскольку он поддерживается в различных средах

и в глобальной сети Internet. Он обладает высокой надёжностью при маршрутизации и поддерживается практически всеми производителями оборудования.

Удалённые пользователи (используя протоколы PPP или ISDN) получают доступ к корпоративной сети таким же образом, как если бы они осуществляли доступ непосредственно к корпоративному шлюзу (несмотря на то, что физически их соединение удалённого доступа осуществляется через сервер Internet-провайдера). На рис. 9.4 показана топологическая картина того, как были бы реализованы эти соглашения службой виртуального удалённого доступа.

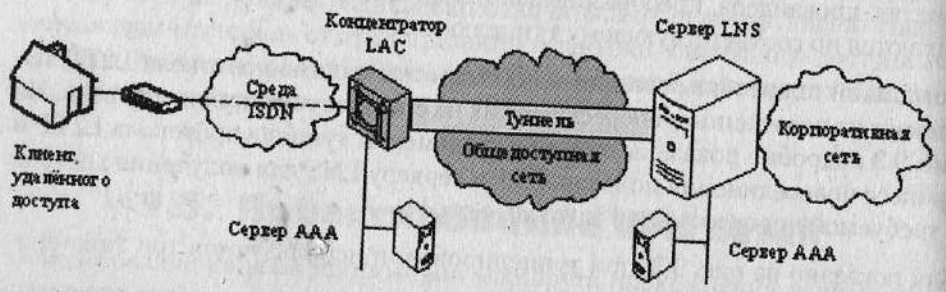


Рис. 9.4. При установке PPP-соединения удаленного пользователя с корпоративной сетью создается полная топология виртуального удаленного доступа

Для безопасного использования этой службы ни на удалённой конечной системе, ни на её корпоративных узлах не требуется специального программного обеспечения.

Аутентификация осуществляется с использованием протокола удалённого доступа PPP. Эта аутентификация может осуществляться пользователем независимо от Internet-провайдера.

Управление адресацией осуществляется так же, как и при использовании выделенных линий, т.е. назначение адреса выполняется компанией пользователя, а не Internet-провайдером.

Авторизацией управляют удалённые пользователи компании. Учёт трафика осуществляется как Internet-провайдером (для регистрации расходов), так и пользователем.

Выполнение этих требований обеспечивается в первую очередь набором функций, которые предоставляются пользователю с помощью туннеля к корпоративной сети по протоколу L2TP. В случае использования протокола PPP, все согласования параметров протокола управления каналом и протокола управления сетью происходят в корпоративной сети пользователя.

Протокол PPP можно использовать для передачи потоков данных от удалённого пользователя до конечной точки, которой является корпоративный шлюз.

Виртуальные частные сети VPN позволяют пользователям и мобильным работникам подсоединяться к корпоративным интранет- и экстранет-сетям. Такие сети позволяют пользователям подсоединяться к Internet локально и осуществлять передачу данных по туннелю в обратном направлении, для подсоединения к корпоративным ресурсам. Это не только уменьшает расходы, связанные с передачей служебной информации, но на также повышает уровень гибкости и масштабируемости сети.

Традиционные сетевые службы удалённого доступа поддерживают только зарегистрированные IP-адреса, что сужает круг приложений, которые могут быть реализованы по виртуальным частным сетям. Протокол L2TP поддерживает несколько протоколов, а также незарегистрированные исправляемые частным образом IP-адреса при работе в Internet. Это позволяет использовать уже существующую инфраструктуру доступа, такую как Internet, модемы, серверы доступа и терминальные адаптеры. Он также позволяет пользователям осуществлять внешнюю поддержку удалённого доступа, тем самым уменьшая служебную нагрузку, расходы на поддержку аппаратного обеспечения, а также даёт возможность концентрировать ресурсы корпоративного шлюза.

9.4. Процесс виртуального удалённого доступа

На рис. 9.5 представлено соединение частной виртуальной сети удалённого доступа VPN с удалённым пользователем, концентратором доступа LAC в точке присутствия (POP) Internet-провайдера и сервером LNS, в домашней локальной сети с использованием туннеля протокола L2TP. Далее описано поэтапное развитие процесса осуществления виртуального удалённого доступа на всём протяжении маршрута следования данных.

1. Удалённый пользователь инициирует соединение протокола PPP с Internet-провайдером, используя аналоговую телефонную сеть или сеть ISDN.
2. Концентратор доступа LAC Internet провайдера принимает соединение в точке присутствия POP и устанавливает канал протокола PPP.
3. После того, как конечный пользователь и сервер LNS согласуют параметры протокола, концентратор LAC частично аутентифициру-

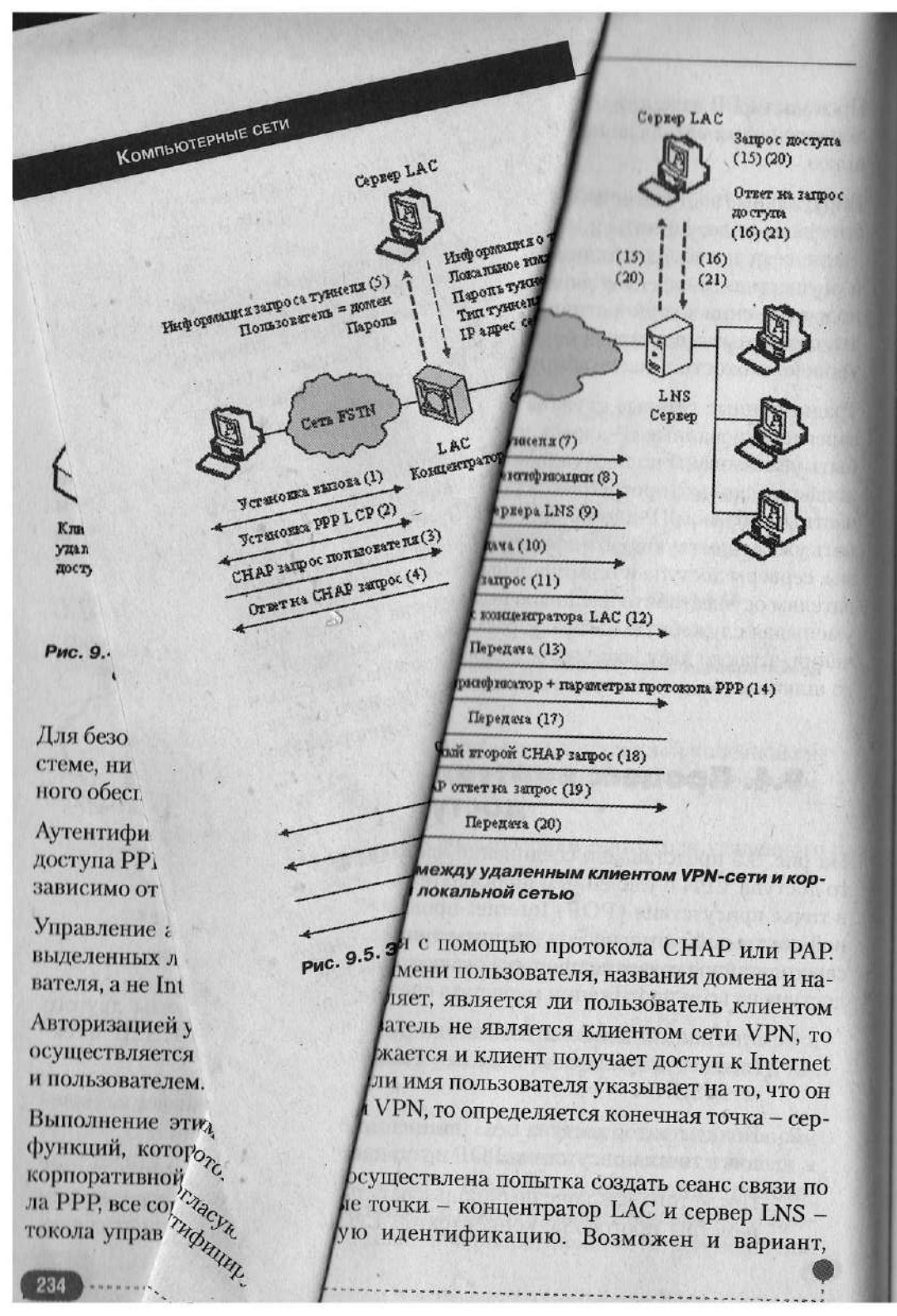


Рис. 9.

Для безо
стеме, ни
ного обес

Аутентифи
доступа PP
зависимо от

Управление
выделенных
вателя, а не Int

Авторизацией
осуществляется
и пользователем.

Выполнение эти
функций, кото
корпоративной
ла PPP, все со
токола управ

между удаленным клиентом VPN-сети и кор-
локальной сетью

Рис. 9.5. З

и с помощью протокола CHAP или PAP.
мени пользователя, названия домена и на-
яет, является ли пользователь клиентом
атель не является клиентом сети VPN, то
жается и клиент получает доступ к Internet
ли имя пользователя указывает на то, что он
VPN, то определяется конечная точка – сер-

осуществлена попытка создать сеанс связи по
ые точки – концентратор LAC и сервер LNS –
ую идентификацию. Возможен и вариант,

когда сервер LNS создает туннель без аутентификации концентратора LAC.

5. После того, как создан туннель, устанавливается сеанс протокола L2TP для конечного пользователя.
6. Концентратор LAC посылает согласованные опции протокола LCP и информацию о частичной аутентификации протокола CHAP или PAP серверу LNS. Сервер LNS передаёт согласованные опции и информацию об аутентификации непосредственно интерфейсу виртуального доступа. Если сконфигурированные опции на виртуальном интерфейсе не совпадают с опциями, согласованными с концентратором LAC, то соединение прерывается и концентратору LAC посылается сообщение о разъединении. Конечным результатом этого является то, что процесс обмена информацией между клиентом удалённого доступа и удалённым сервером LNS выглядит как эксклюзивный, как если бы в нём не участвовали промежуточные устройства (концентратор LAC). На рис. 9.5 наглядно показана L2TP последовательность входного вызова.

Особые свойства службы виртуального удалённого доступа

Аутентификация и безопасность сети. В традиционном сценарии удалённого доступа Internet провайдер, использующий сервер безопасности, выполняет аутентификацию, запрашивая у пользователя удалённого доступа его имя и пароль. Если удаленный пользователь проходит такую проверку, то начинается этап авторизации.

Для службы виртуального удалённого доступа Internet провайдер выполняет аутентификацию с целью выяснить идентификационные параметры пользователя и, как следствие этого, желательный для пользователя корпоративный шлюз. На этом этапе соответствие паролей не проверяется. Сразу после того, как определён корпоративный шлюз, инициируется соединение с использованием собранной Internet-провайдером информации об аутентификации. Корпоративный шлюз заканчивает процесс аутентификации, принимая или отвергая это соединение. Например, в запросе протокола PAP соединение отвергается, если имя пользователя или его пароль оказываются неправильными. В случае, когда соединение принято, корпоративный шлюз может выполнить следующую фазу аутентификации на уровне протокола PPP.

Для каждого установленного туннеля протокола L2TP служба безопасности этого протокола генерирует случайным образом уникальный ключ, предна-

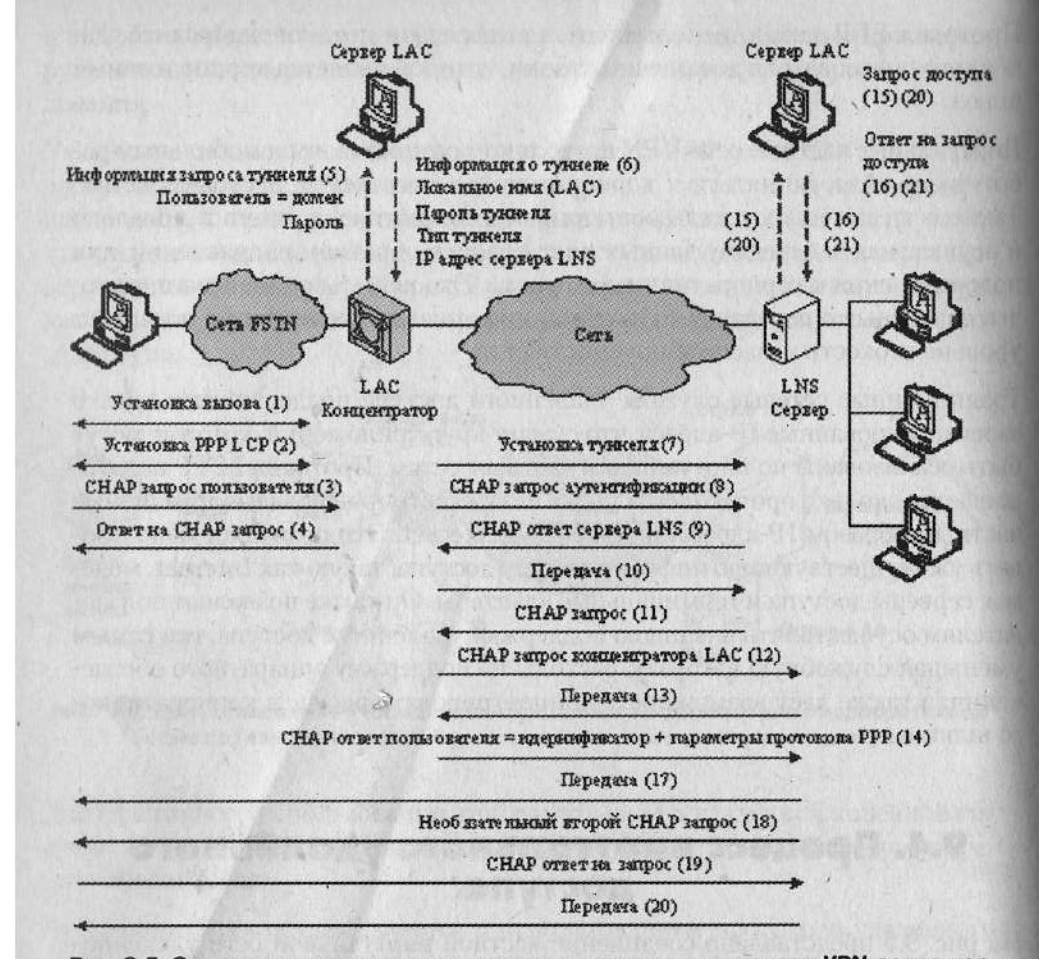


Рис. 9.5. Этапы установки соединения между удаленным клиентом VPN-сети и корпоративной локальной сетью

ет конечного пользователя с помощью протокола CHAP или PAP. Служба идентификации имени пользователя, названия домена и набранного номера определяет, является ли пользователь клиентом сети VPN. Если пользователь не является клиентом сети VPN, то аутентификация продолжается и клиент получает доступ к Internet или к другой службе. Если имя пользователя указывает на то, что он является клиентом сети VPN, то определяется конечная точка – сервер LNS.

4. Перед тем, как будет осуществлена попытка создать сеанс связи по туннелю, его конечные точки – концентратор LAC и сервер LNS – выполняют взаимную идентификацию. Возможен и вариант, когда сервер LNS создает туннель без аутентификации концентратора LAC.

5. После того, как создан туннель, устанавливается сеанс протокола L2TP для конечного пользователя.
6. Концентратор LAC посылает согласованные опции протокола LCP и информацию о частичной аутентификации протокола CHAP или PAP серверу LNS. Сервер LNS передаёт согласованные опции и информацию об аутентификации непосредственно интерфейсу виртуального доступа. Если сконфигурированные опции на виртуальном интерфейсе не совпадают с опциями, согласованными с концентратором LAC, то соединение прерывается и концентратору LAC посылается сообщение о разъединении. Конечным результатом этого является то, что процесс обмена информацией между клиентом удалённого доступа и удалённым сервером LNS выглядит как эксклюзивный, как если бы в нём не участвовали промежуточные устройства (концентратор LAC). На рис. 9.5 наглядно показана L2TP последовательность входного вызова.

Особые свойства службы виртуального удалённого доступа

Аутентификация и безопасность сети. В традиционном сценарии удалённого доступа Internet провайдер, использующий сервер безопасности, выполняет аутентификацию, запрашивая у пользователя удалённого доступа его имя и пароль. Если удаленный пользователь проходит такую проверку, то начинается этап авторизации.

Для службы виртуального удалённого доступа Internet провайдер выполняет аутентификацию с целью выяснить идентификационные параметры пользователя и, как следствие этого, желательный для пользователя корпоративный шлюз. На этом этапе соответствие паролей не проверяется. Сразу после того, как определён корпоративный шлюз, инициируется соединение с использованием собранной Internet-провайдером информации об аутентификации. Корпоративный шлюз заканчивает процесс аутентификации, принимая или отвергая это соединение. Например, в запросе протокола PAP соединение отвергается, если имя пользователя или его пароль оказываются неправильными. В случае, когда соединение принято, корпоративный шлюз может выполнить следующую фазу аутентификации на уровне протокола PPP.

Для каждого установленного туннеля протокола L2TP служба безопасности этого протокола генерирует случайным образом уникальный ключ, предна-

значенный для защиты. Внутри туннеля L2TP каждый мультиплексированный сеанс поддерживает номер последовательности для предотвращения дублирования пакетов. Аппаратное и программное обеспечение фирмы Cisco предоставляет значительную гибкость, позволяя пользователю осуществлять сжатие данных на стороне клиента. Кроме того, шифрование в туннеле может осуществляться с использованием протокола IPSec.

Авторизация. При предоставлении традиционной службы удалённого доступа Internet-провайдер поддерживает отдельные профили для каждого пользователя, которые определяют авторизацию. На основе этого сервер безопасности, взаимодействуя с сервером сетевого доступа, предоставляет доступ на основе политик использования и подсоединения пользователей. Характер этих политик может меняться от простых фильтров по источнику/получателю для небольшой группы сайтов до сложных алгоритмов, учитывающих конкретный характер используемых приложений, время дня, в которое происходит удалённый доступ, а также длинные списки разрешённых или, наоборот, запрещённых пунктов назначения. Этот процесс может потребовать значительных вычислительных ресурсов от Internet-провайдера, особенно если ему требуется обеспечить доступ удалённых пользователей от имени компаний, которым требуется постоянно изменять свою политику.

При использовании службы виртуального удалённого доступа вся тяжесть подробной авторизации, основанной на политиках, возлагается непосредственно на компанию удалённого пользователя. В условиях, когда между удалёнными пользователями и их корпоративным шлюзом устанавливается сквозное соединение, все операции по авторизации могут быть выполнены так, как если бы удалённые пользователи осуществляли непосредственный доступ к своей корпоративной сети. Такой подход освобождает Internet-провайдера от необходимости поддержания большой базы данных профилей индивидуальных пользователей многих компаний. При этом служба виртуального удалённого доступа становится более безопасной для компаний. Служба виртуального удалённого доступа также позволяет компаниям быстро реагировать на изменения состава групп своих удалённых пользователей.

Выделение адресов. При использовании традиционной службы удалённого доступа к Internet пользователю IP-адрес выделяется динамически из набора доступных адресов провайдера. В этом случае удалённые пользователи имеют ограниченный доступ к ресурсам своей корпоративной сети или вообще его не имеют, поскольку брандмауэры и политики безопасности запрещают доступ с внешних IP-адресов к корпоративной сети.

При использовании службы виртуального удалённого доступа корпоративный шлюз может выделять адреса, которые являются внутренними адресами

корпоративной сети (и могут быть адресами, зарезервированными для частных сетей или не быть IP-адресами). Поскольку туннели протокола L2TP оперируют исключительно на уровне преобразованных (инкапсулированных) кадров, реальные политики управления адресами для корректировки службы виртуального удалённого доступа для всех целей обработки протокола PPP для пользователя удалённого доступа выглядят, как будто он соединён непосредственно с корпоративным шлюзом.

Учёт. Сервер сетевого доступа и корпоративный шлюз могут подсчитывать количество пактов, октетов и количество созданных и разорванных соединений.

Поскольку виртуальный удалённый доступ является службой доступа, учёт попыток доступа (особенно неудачных) представляет значительный интерес. Корпоративный шлюз может отвергнуть попытки создания новых соединений на основе информации аутентификации, собранной Internet-провайдером, с внесением соответствующих записей в файл учёта. Возможны случаи, когда корпоративный шлюз принимает соединение и продолжает аутентификацию, однако впоследствии отсоединяет клиента. В таких случаях сообщение об отсоединении, направляемое в обратном направлении – Internet-провайдеру, может также включать в себя указание причины отсоединения. Поскольку корпоративный шлюз может отказать в создании соединения, основываясь на информации, собранной Internet-провайдером, такой учёт позволяет легко отличить серию неудачных попыток создания соединения от серии коротких успешных соединений. Без этого корпоративному шлюзу приходится всегда принимать запросы на соединения и обмениваться многочисленными PPP-пакетами с удалённой системой.

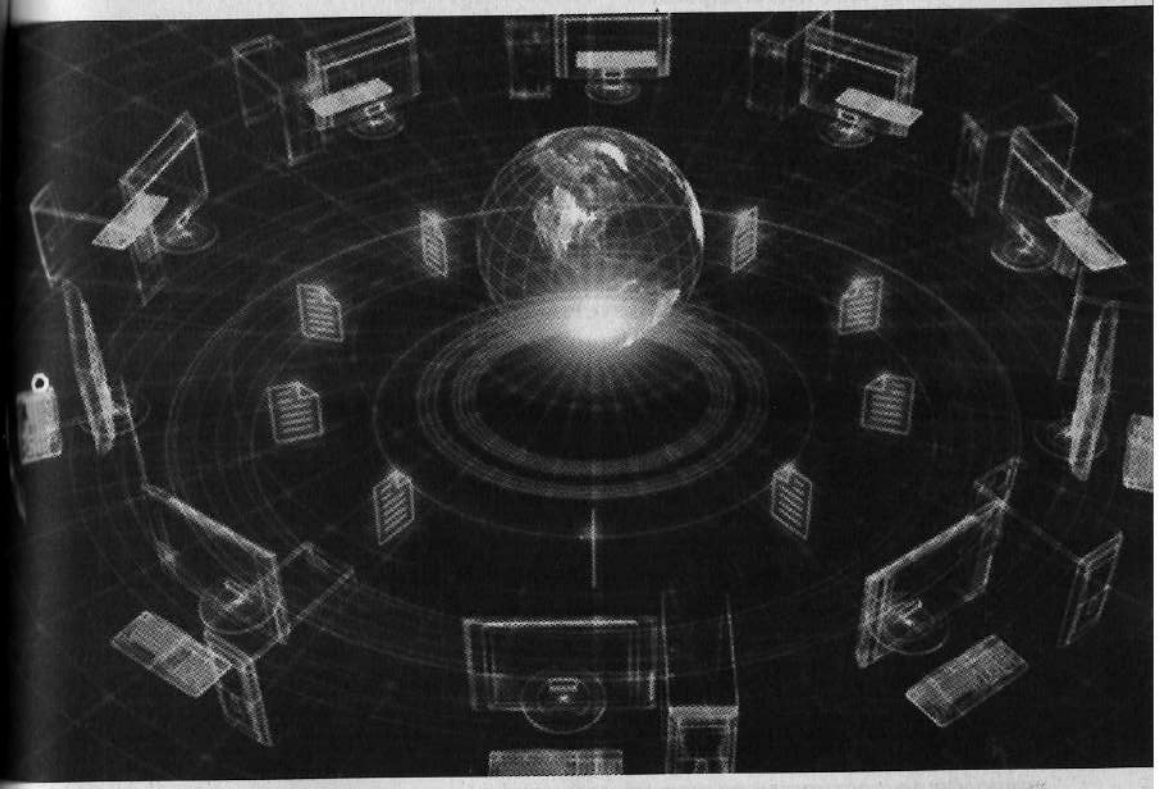
Контрольные вопросы

1. *Объясните назначение политики безопасности.*
2. *Назовите основные функции политики безопасности.*
3. *В чём заключается политика безопасности на магистральном уровне?*
4. *В чём заключается политика безопасности на уровне распределения?*
5. *Объясните назначение списков доступа.*
6. *Каким образом используются списки доступа?*
7. *Какие типы списков доступа Вам известны?*
8. *В чём заключается политика безопасности на уровне доступа?*
9. *Поясните назначение уровней привилегий.*

10. Каким образом обеспечивается физическая безопасность устройств сети?
11. В чём заключается безопасность порта?
12. В чём заключается управление коммутатором?
13. Объясните назначение частных виртуальных сетей.
14. Какие типы частных виртуальных сетей Вам известны?
15. Какую последовательность шагов необходимо выполнить для реализации частных виртуальных сетей?
16. В чём заключается процесс проектирования виртуальных частных сетей?
17. Объясните назначения туннеля в VPN.
18. Какие Вам известны протоколы туннелирования в VPN?
19. Поясните процесс виртуального удалённого доступа.
20. Объясните основные шаги, выполняемые для организации виртуального удалённого доступа.
21. Какие свойства службы виртуального удалённого доступа Вам известны?

Глава 10.

СИСТЕМА АДРЕСАЦИИ В IP-СЕТЯХ



Несмотря на большое и всё увеличивающееся количество функций, выполняемых коммутаторами, они имеют ряд существенных недостатков, не позволяющих на их основе строить глобальные сети с большим количеством альтернативных путей передачи данных. Основные недостатки коммутаторов заключаются в следующем:

- коммутатор работает только в рамках одной сети;
- коммутаторы не имеют возможности защитить сеть от широковещательного шторма;
- в коммутаторах сложно решается проблема управления трафиком;
- коммутаторы работают с недостаточно гибкой системой адресации на основе MAC-адреса;
- коммутаторы не поддерживают трансляцию протоколов.

Сетевой уровень модели OSI предназначен для координации работы подсетей, лежащих на пути следования пакета. Для эффективного управления передачей пакета по составным сетям сетевому уровню требуется собственная система адресации, не зависящая от способов адресации узлов в подсетях, но при этом позволяющая однозначно идентифицировать любой узел составной сети.

В современных IP-сетях используется трёхуровневая система адресации: локальный MAC-адрес, IP-адрес и символьный адрес.

Для передачи информации внутри сети применяется локальный физический MAC-адрес, который однозначно идентифицирует (определяет) узел сети. Внутри сетей информация передаётся только по MAC-адресу, структура которого была рассмотрена подробно в предыдущих главах.

10.1. IP-адрес

Для передачи информации между сетями сетевой уровень использует свою собственную логическую систему адресации, которая тоже позволяет однозначно идентифицировать узел в составной сети, но только на логическом уровне. IP-адрес представляет собой 32-х разрядное (четырёхбайтное) двоичное число, состоящее из двух частей: старшая часть адреса определяет номер сети, а младшая – номер узла в этой сети, как показано на рис. 10.1.

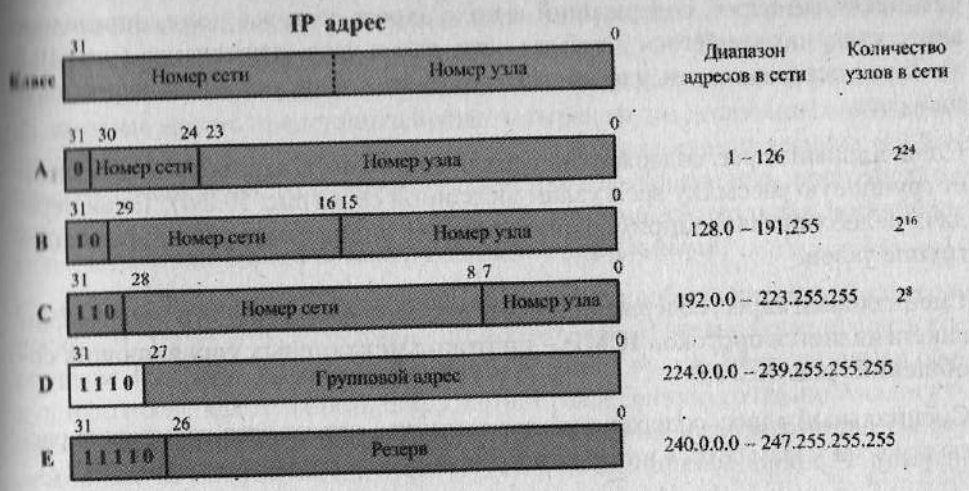


Рис. 10.1. Классы IP-адресов

Всё адресное пространство разделено на классы, признаком класса является наличие определённой комбинации в старших разрядах адреса. Всего определено 5 классов, которые обозначаются первыми буквами латинского алфавита от A до E.

Признаком класса A является наличие ноля в старшем разряде адреса. При этом под номер сети отводится старший байт, а под номер узла в этой сети – младшие три байта, поэтому класс A определяет до 126 сетей по 2^{24} узлов в каждой.

Признаком класса B является наличие комбинации 10 в старших разрядах адреса, что позволяет иметь до 49 150 сетей по 2^{16} узлов в каждой.

Признаком класса C является комбинация 110 в старших разрядах адреса, что позволяет иметь до 7 340 030 сеть по 2^{8} узлов в каждой.

Признаком класса D является наличие комбинации 1110 в старших разрядах адреса. Адреса класса D являются групповыми, то есть информация рассылается всем узлам, которые имеют указанные 28 младших разрядов.

Признаком класса E является наличие в старших разрядах адреса комбинации 11110. Эти адреса в количестве чуть больше 2-х миллиардов зарезервированы для специальных нужд и для адресации узлов не используются.

Кроме основных пяти классов в системе IP-адресов определены ещё несколько специальных адресов (рис. 10.2).

Специальный адрес, содержащий в поле адреса сети все ноли, определяет адрес узла, находящегося в той же сети, что и узел-отправитель (рис. 10.2 а). Такая передача очень удобна, поскольку можно не указывать адрес текущей сети.

Специальный адрес, содержащий все единицы в поле адреса узла, определяет групповую рассылку всем узлам указанной сети (рис. 10.2 *б*). Такая передача целесообразна, например, при передаче видео- или аудиоинформации группе узлов.

Специальный адрес, содержащий все ноли, сообщает о том, что генератором пакета является протокол **ICMP – протокол межсетевых управляющих сообщений** (рис. 10.2 *в*).

Специальный адрес, содержащий все единицы, определяет групповую рассылку всем узлам сети, в которой находится узел-отправитель (рис. 10.2 *г*).

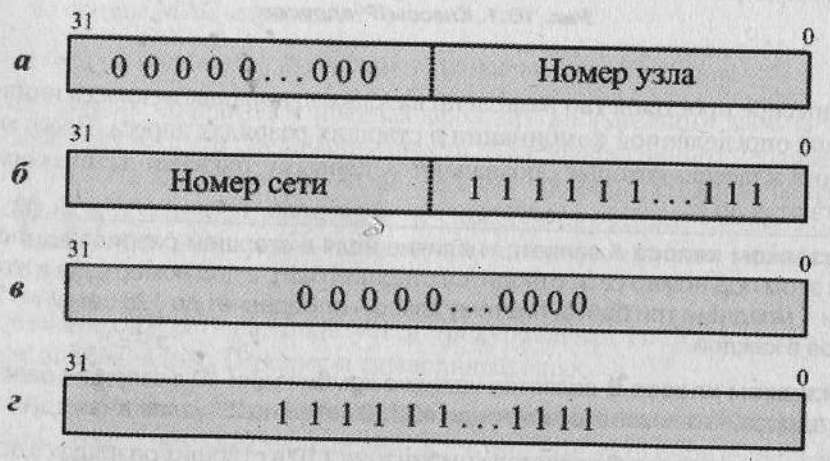


Рис. 10.2. Специальные IP-адреса:

Специальный адрес 127.0.0.1 зарезервирован для организации обратной связи при тестировании программного обеспечения. Это необходимо для того, чтобы при отладке ПО пакеты не засоряли сеть. В этом случае сетевая карта просто не передаёт пакеты с таким адресом за пределы компьютера в сеть.

а) узел получателя находится в сети узла отправителя;

б) рассылка всем узлам указанной сети;

в) генератором пакета является протокол ICMP;

г) рассылка всем узлам сети, в которой находится узел-отправитель.

Наличие специальных и резервных адресов приводит к уменьшению диапазона доступных адресов до величины примерно 2^{31}. Поскольку все выше перечисленные адреса представляют собой многоразрядные двоичные числа, которые можно представить в виде четырёх групп десятичных чисел (по числу байт), разделённых точками, не вызывающих никаких ассоциаций, например 192.44.39.16, то пользователю их сложно запоминать, поэтому каждому логическому адресу ставится в соответствие символьная последовательность. Такие адреса принято называть символьными.

Символьные адреса в использовании намного удобнее числовых адресов, поскольку могут отражать имена пользователей. В этом случае послать сообщение Гале или Маше по их именам намного проще, нежели указав бездушную последовательность цифр, хотя и разделённую точками. Аналогичная система адресации используется при указании пути к файлам, расположенным на некотором носителе в одной из вложенных папок. Например, приведённый ниже путь:

C:\Документ\Admin\Меню\Программы\Стандартные\Служебные\Apr.doc

указывает на то, что искомый текстовый документ Apr.doc расположен на носителе С в папке Документ, которая имеет вложенную папку Admin, которая в свою очередь содержит вложенную папку Программы, в которой находится папка Стандартные, в которой находится папка Служебные, а уж в ней находится документ. Подобная иерархическая структура позволяет очень просто хранить большое количество документов, а самое главное, обеспечивает быстрый их поиск, вследствие того, что они сгруппированы по смыслу в группы большего размера.

В системе IP-адресов применяется аналогичная система, только её запись происходит в обратном порядке, то есть сначала указывается имя документа, а затем последовательно названия всех вложенных папок и имя носителя. Так, вышеприведённая запись в системе IP-адресов будет выглядеть следующим образом:

Apr.Служебные.Стандартные.Программы.Меню.Admin.Документ.C

Нетрудно заметить, что вместо разделительной обратной косой черты используется точка. Для передачи информации в IP-сетях используется только 32-х разрядный двоичный IP-адрес, поэтому необходимо преобразовать символьный адрес в двоичный. Для этих целей необходимо иметь таблицу соответствия всех символьных имён двоичным адресам. Однако размер таблицы может достигать более 2-х миллиардов строк. Хранение таблицы соответствия в одном месте нецелесообразно, поскольку все пользователи символьных имён должны будут сначала обратиться к таблице соответствия

для получения IP-адреса, а уже затем производить передачу информации по полученному адресу. Кроме этого, для управления таблицей (присвоение имён пользователям, удаление пользователя, модификация записи и так далее) доступ к таблице должны иметь администраторы. Поэтому при большом количестве обращений в единицу времени для получения приемлемого времени ответа может потребоваться машина с громадным быстродействием, выход из строя которой приведёт к краху всей сети.

Для уменьшения нагрузки на такую ЭВМ целесообразно применять децентрализованный метод, то есть разделить таблицу на части и каждую отдельную часть хранить на отдельной машине. В результате количество записей на одной машине резко уменьшается, управление таблицей упрощается и потеря некоторой части общей таблицы в результате выхода из строя ЭВМ не приведёт к выходу всей сети из строя.

Совокупность имён, у которых старшая часть совпадает, образует **домен имён**. Дерево имён начинается с корня и может содержать произвольное количество составных частей (рис. 10.3). Разделение имени на части позволяет значительно упростить процесс управления именами и снизить нагрузку на ЭВМ имён домена.

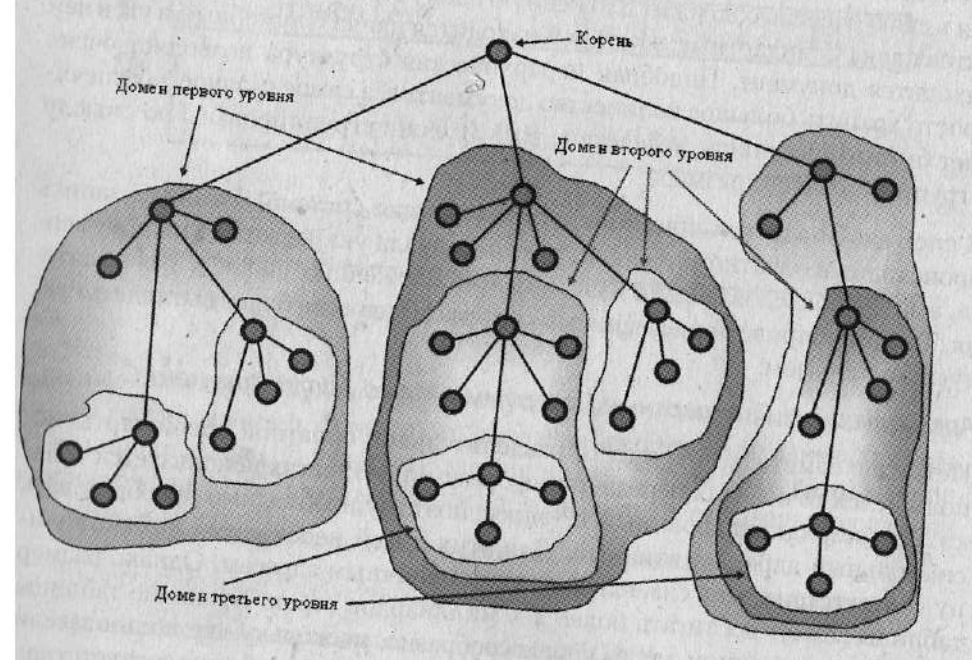


Рис. 10.3. Доменная система в IP-сетях

Соответствие между доменными именами и IP-адресами реализуется средствами – системы доменных имён DNS. DNS представляет собой централизованную службу, основанную на распределённой базе данных типа клиент-сервер. Сервер домена хранит все имена своего домена и ссылки (адреса) к серверам своих подчинённых доменов и к серверу вышестоящего домена. Эти ссылки связывают все DNS-серверы в единую службу системы доменных имён. При этом процедура поиска адреса по символьному имени заключается в последовательном просмотре всей цепочки подчинённых доменов, входящих в имя, начиная с корневого домена.

Когда пользователь обращается к DNS-серверу своего домена за помощью в определении IP-адреса по символьному имени, то DNS-сервер может сразу возвратить искомый адрес. Правда, это возможно только в случае, если запрошенное имя входит в тот же домен, что и имя пользователя, или когда сервер уже узнал указанное имя ранее и сохранил его в своём буфере.

Если сервер домена не знает ответ, то он выполняет запрос к своему вышестоящему серверу DNS, а тот в свою очередь может обратиться тоже к вышестоящему DNS-серверу, и так можно дойти до корневого сервера DNS. Корневой сервер DNS в соответствии с доменным именем производит обращение к соответствующему подчинённому домену, а тот в свою очередь к своему соответствующему подчинённому домену и так далее, до тех пор, пока не будет достигнут нужный сервер домена. Ответ на обработанный запрос проходит путь в обратном направлении, пока не достигнет клиента, пославшего запрос. При этом каждый сервер DNS запоминает полученные ответы в своём буфере на определённое время – обычно от нескольких часов до нескольких дней. Такой приём позволяет значительно сократить время поиска нужной информации в системе.

Для управления и назначения корневых доменов в сети Интернет в 1972 году в США был создан информационный сетевой центр Интернета InterNIC (Internet's Network Information Center). В его функции входило назначение доменов верхнего уровня для каждой страны. Для обозначения страны используются двух- или трёхбуквенные обозначения в соответствии с международным стандартом ISO 3166.

Однако деятельность центра InterNIC, монополизировавшего регистрацию имён в доменах общего пользования за достаточно высокую цену, стала тормозить рост количества зарегистрированных доменных имён. Например, на начало 1998 года в мире насчитывалось всего три миллиона зарегистрированных доменных имён и доменов общего пользования было определено всего три: com, net и org. Поэтому в 1998 году для регулирования вопросов, связанных с доменными именами, IP-адресами и прочими аспектами функционирования Интернета, была создана международная некоммерческая

организация ICANN (Internet Corporation for Assigned Names and Numbers), которая основана на принципе свободного доступа аккредитованных регистраторов к реестрам доменных имён. На сегодняшний момент в доменных зонах общего пользования работает более 900 аккредитованных регистраторов, благодаря чему количество зарегистрированных доменов уже превысило 160 миллионов.

В рамках развития системы адресации корпорация ICANN последовательно расширяет список доменов общего пользования, в том числе с использованием символов национальных языков. Каждый домен управляется отдельным аккредитованным регистратором, который обычно делит его на подчинённые домены и передаёт функции управления этими подчинёнными доменами другим организациям. В России аккредитованным регистратором ICANN является компания является Региональный сетевой информационный центр (RU-CENTER), отвечающий за создание подчинённых доменов в доменах ru и рф.

10.2. Структура IP-пакета

IP-пакет состоит из заголовка и поля данных. Весь пакет представляется в виде набора 32-х разрядных слов, как показано на рис. 10.4. Самое первое поле в заголовке пакета – поле номера версии, оно занимает 4 бита. В настоящее время используется версия IPv4, однако уже начался переход на новую версию IPv6. Версия пакета определят количество и размер полей заголовка, а также количество бит, отводимое под поле адреса. Версия IPv4 определяет размер поля адреса в 32 бита.

Поле "длина заголовка" занимает 4 бита и определяет размер заголовка в 32-х разрядных словах. Размер заголовка может колебаться в пределах от 5 до 15 слов, в зависимости от количества дополнительной служебной информации.

Поле «тип сервиса» занимает 8 бит и определяет критерии выбора маршрута и скорость обработки пакета. Поле приоритета определяет важность пакета и занимает 3 бита, при этом более важные пакеты должны обрабатываться в первую очередь. Признаки D, T, и R определяют критерии выбора маршрута: минимальные задержки, максимальная пропускная способность и максимальная надёжность доставки соответственно. Во многих случаях улучшение одного из указанных параметров связано с ухудшением других, поэтому в реальных условиях нецелесообразно устанавливать более двух критериев одновременно. Кроме этого, обработка каждого признака связана с дополнительными вычислениями, что приводит к необходимости иметь более

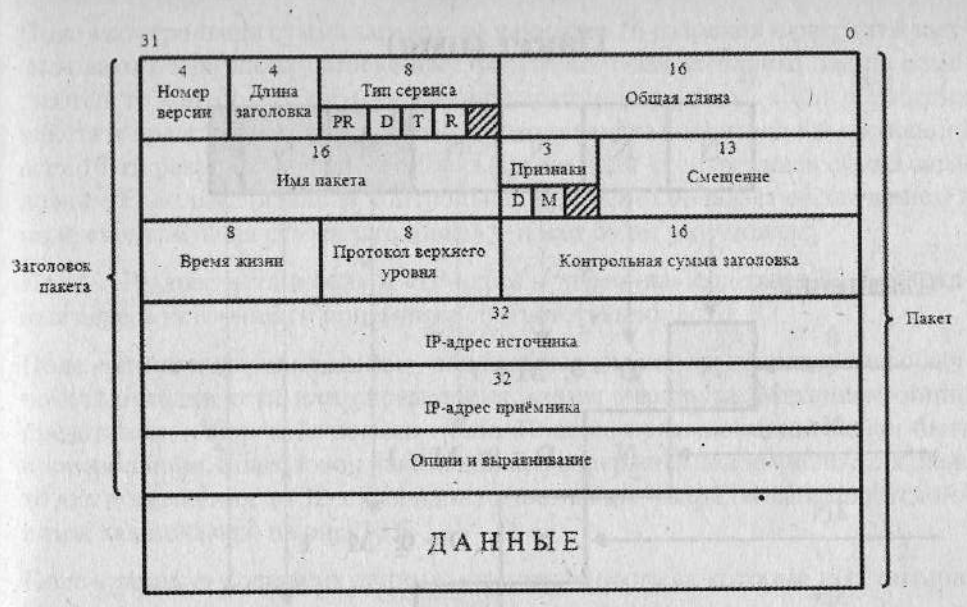


Рис. 10.4. Структура IP-пакета IPv4

высокопроизводительные процессоры в маршрутизаторах. Неиспользуемые биты имеют значение ноль.

Поле «общая длина» пакета занимает 16 разрядов и определяет размер всего пакета в байтах. Поэтому максимальный размер пакета может достигать 64 Кбайт.

Поле «имя пакета» занимает 16 разрядов и используется только в случае деления пакетов на части, при этом все фрагменты должны иметь одинаковое имя.

Поле «признаки» занимает 3 бита и определяет возможности фрагментации. Признак D установленный в ноль – разрешает фрагментацию, а признак M установленный в единицу – определяет фрагмент как промежуточный, но не последний.

Поле «смещение» определяет смещение поля данных данного фрагмента относительно начала поля данных исходного пакета. Оно необходимо для правильной сборки пакета из фрагментов и должно быть кратно 8 байтам. На рис. 10.5 приведён пример деления пакета на части, принцип определения смещения и значения признаков. При этом все фрагменты должны иметь одно имя.

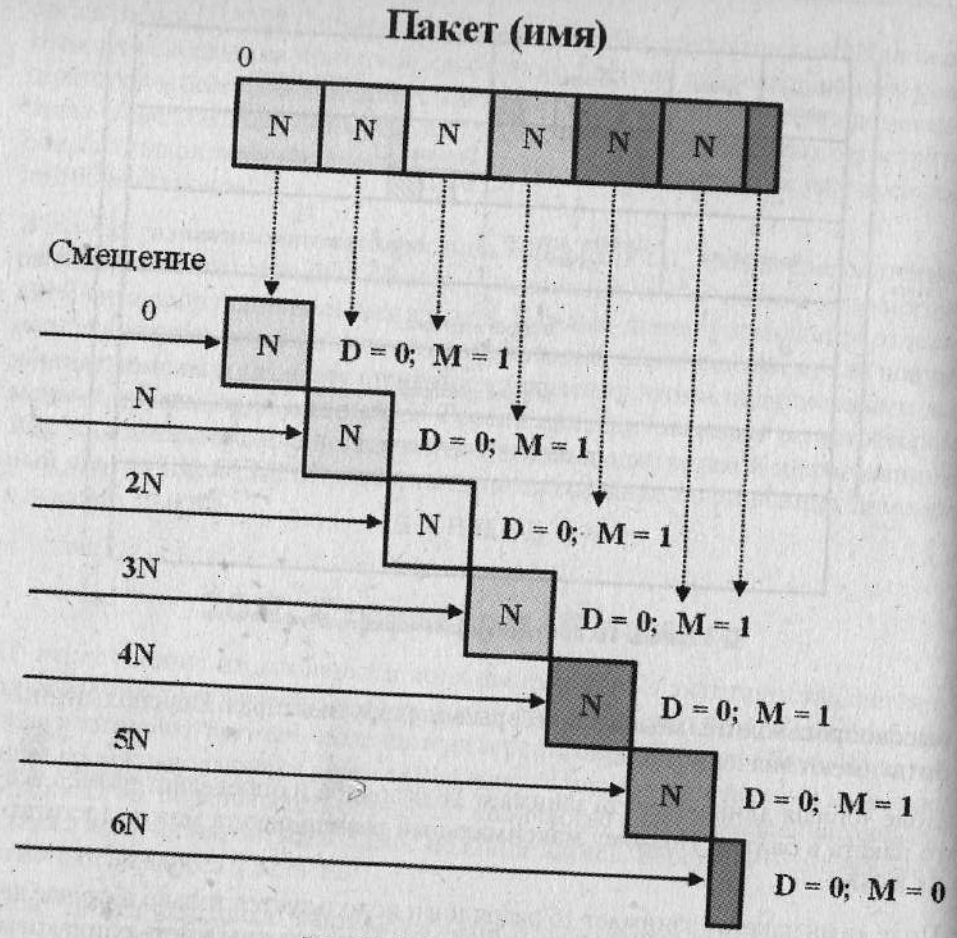


Рис. 10.5. Фрагментация пакета

Поле «время жизни» занимает 8 бит и определяет максимальное время существования пакета в сети. Время жизни измеряется в секундах и определяется источником. В каждом маршрутизаторе, после обработки пакета, время жизни уменьшается на величину округлённую в большую сторону. Например, если время обработки составило 1,001 секунды, то время жизни будет уменьшено на 2. Поскольку современные маршрутизаторы практически никогда не обрабатывают пакеты дольше секунды, то время жизни можно рассматривать как максимальное количество маршрутизаторов, которые может пройти пакет. Если время жизни станет нулевым, до того как пакет достигнет места назначения, то он будет уничтожен.

Поле «протокол верхнего уровня» занимает 8 разрядов и определяет тип протокола верхнего уровня, которому принадлежат данные, размещённые в поле данных.

Поле «контрольная сумма заголовка» занимает 16 разрядов и рассчитывается только по заголовку. Поскольку некоторые поля заголовка пакета изменяются, то контрольная сумма рассчитывается каждый раз после обработки пакета и представляет собой сумму по модулю два (поразрядное сложение) всех 16-ти разрядных слов заголовка, кроме поля «контрольная сумма заголовка». Если рассчитанная контрольная сумма не совпадает со значением в поле «контрольная сумма заголовка» – пакет будет уничтожен.

Поля «IP-адрес источника» и «IP-адрес приёмника» содержат 32-х разрядные адреса источника и приёмника соответственно.

Поле «опции и выравнивание» – необязательное поле и используется обычно для отладки сети или определения длины маршрута. Механизм опций представлен набором из восьми типов. Поскольку число опций может быть произвольным, а заголовок пакета должен содержать целое число 32-х слов, то для дополнения до 32-х разрядов недостающие разряды заполняются нолями, как показано на рис. 10.6.

Поле «данные» содержит данные, ответственность за которые несёт отправитель.

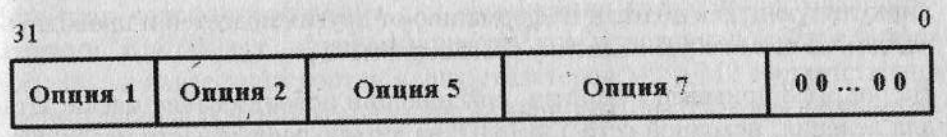


Рис. 10.6. Опции и выравнивание

10.3. Маски постоянной длины

Система адресации, основанная на использовании классовых адресов, имеет целый ряд существенных недостатков, связанных с совершенно не оптимальным использованием адресного пространства. Как было отмечено ранее, классовые сети могут иметь в своём составе фиксированное количество адресов: 2^8, 2^{16} или 2^{24}. Однако если администратору требуется другое количество адресов, например 2013, то 256 адресов явно мало, а 65536 – крайне много. Для деления адресного пространства на произвольное количество адресов и был придуман механизм масок.

Маска представляет собой 32-х разрядное двоичное число, старшие разряды которого имеют значение единица, а младшие – равны нулю. Количество единиц определяет количество разрядов в адресе, используемое для определения адреса сети. На рис. 10.7 приведён пример, каким образом с помощью

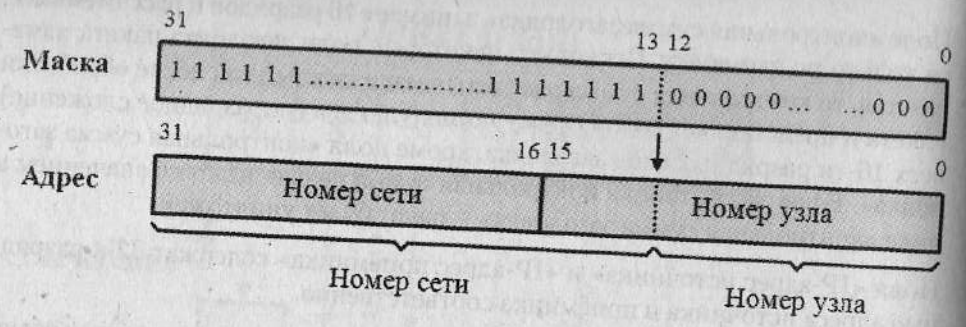


Рис. 10.7. Использование маски

маски происходит перераспределение количества разрядов между адресом сети и адресом узла. Таким образом, маска определяет количество разрядов адреса, интерпретируемое под номер сети.

Допустим, компания получила в своё распоряжение адрес класса B: 138.78.0.0. Но администратору таким количеством адресов сложно управлять. Тем более что в компании есть три независимых отделения и выделение для каждого из них отдельной подсети позволит надёжно изолировать циркулирующую в подсети информацию от других подсетей и проводить в каждой из них особую политику безопасности.

На рис. 10.8 приведён механизм, поясняющий использование маски. Итак, для деления исходной сети 138.78.0.0 на четыре равные части необходимо иметь два дополнительных разряда в составе адреса сети. Эти два разряда позаимствованы из старшей части адреса узла исходной сети. Значения двух разрядов позволяют иметь четыре комбинации от 00 до 11, поэтому адреса подсетей будут различаться значением именно этих разрядов. Поскольку они являются старшими разрядами в третьем байте, то адреса подсетей будут иметь следующие значения: X.X.0.0, X.X.64.0, X.X.128.0 и X.X.192.0, где X.X – значения старших разрядов адреса исходной сети. В связи с тем, что количество разрядов для определения адреса подсети увеличилось на 2 и стало вместо 16 – 18, то маска подсети должна содержать в старших 18-ти разрядах единицы. В десятичной записи маска будет иметь следующий вид – 255.255.192.0.

В результате администратор сети получил сеть, структура которой показана на рис. 10.9. Все пакеты из внешней сети поступают через маршрутизатор M1. Для разделения потоков информации во внутренней сети используется дополнительный маршрутизатор M2. Все узлы распределены по трём подсетям, которым были присвоены номера 138.78.0.0, 138.78.64.0 и 138.78.128.0. Все подсети имеют маску одинаковой длины – 255.255.192.0. Каждая

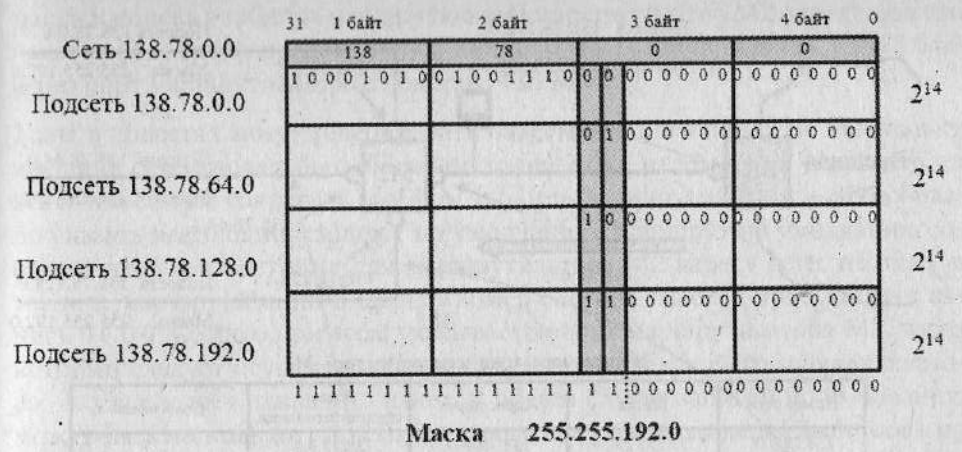


Рис. 10.8. Разделение адресного пространства на четыре равные части

подсеть подключена к соответствующим образом сконфигурированным портам маршрутизатора M2.

Четвёртая подсеть необходима для организации связи между маршрутизаторами. В этой сети, в нашем примере, задействованы всего три адреса узла: два для адресации портов маршрутизаторов M1 и M2 соответственно и один для подключения компьютера администратора. Ещё два адреса узла 138.78.192.0 и 138.78.255.255 являются особыми адресами, поэтому огромное количество узлов, а именно 2^{14} - 5, в этой подсети не используется.

Для внешней сети полученная разделённая сеть выглядит как единая сеть класса B, а для администратора – это полноценная составная сеть, содержащая три отдельные подсети. Деление сети на подсети позволяет скрыть внутреннюю структуру сети от внешнего наблюдателя, что повышает безопасность сети. Поступающие извне пакеты распределяются между этими подсетями маршрутизатором M2 в соответствии с таблицей маршрутизации.

Для того чтобы маршрутизатор мог правильно рассортировать пакеты, для каждой из подсетей в таблице маршрутизации (рис. 10.9) отводится целая строка, в которой указывается номер сети, маска подсети, адрес порта маршрутизатора, на который необходимо передать пакет, чтобы попасть в указанную подсеть. В поле «адрес следующего маршрутизатора» указывается адрес порта следующего маршрутизатора, этот адрес необходим в том случае, когда требуется передать пакет за пределы сети. В поле «служебная информация» обычно указывается время создания записи, доступен ли канал связи и так далее.

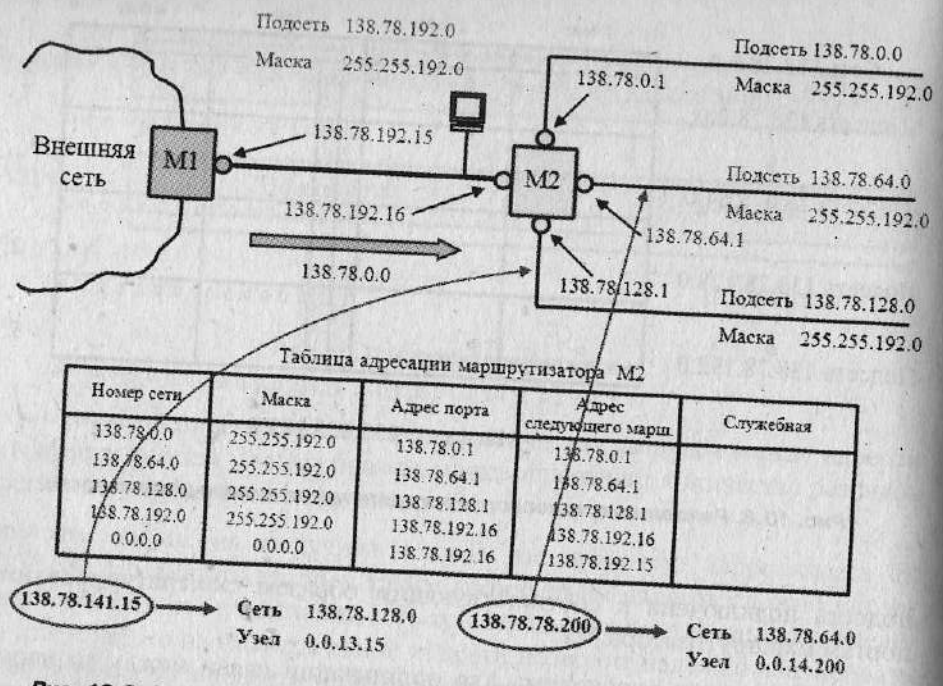


Таблица адресации маршрутизатора M2

Номер сети	Маска	Адрес порта	Адрес следующего марш.	Служебная
138.78.0.0	255.255.192.0	138.78.0.1	138.78.0.1	
138.78.64.0	255.255.192.0	138.78.64.1	138.78.64.1	
138.78.128.0	255.255.192.0	138.78.128.1	138.78.128.1	
138.78.192.0	255.255.192.0	138.78.192.16	138.78.192.16	
0.0.0.0	0.0.0.0	138.78.192.16	138.78.192.15	

Рис. 10.9. Составная сеть, полученная на основе маски постоянной длины

В нашем примере все внешние маршрутизаторы пакеты, начинающиеся на 138.78, определяют как пакеты класса B и направляют их к маршрутизатору M1. Маршрутизатор M1 в свою очередь все пакеты сети 138.78.0.0 транслирует на порт 138.78.192.16 маршрутизатора M2. Маршрутизатор M2 накладывает маску (извлекаемую из таблицы маршрутизации) на полученный адрес и определяет номер подсети. В соответствии с номером подсети он направляет пакет на соответствующий порт маршрутизатора M2.

Допустим, пришёл пакет с адресом 138.78.141.15, маршрутизатор M2 после наложения маски 255.255.192.0 определил, что этот пакет предназначен для подсети 138.78.128.0. В таблице маршрутизации, методом последовательного просмотра строк, маршрутизатор нашёл порт назначения – 138.78.128.1. Поскольку в качестве следующего маршрутизатора указан этот же порт, то маршрутизатор делает вывод о том, что следующего маршрутизатора нет и искомая сеть непосредственно подключена к найденному порту. Следующим шагом маршрутизатора M2 является передача пакета узлу 0.0.13.15 подсети с номером 138.78.128.0.

Рассмотрим ещё один пример. Из внешней сети на порт 138.78.192.16 маршрутизатора M2 поступил пакет с адресом 138.78.78.200. После наложения

маски и поиска в таблице маршрутизации маршрутизатор M2 определил, что пакет необходимо передать узлу с номером 0.0.14.200 в подсеть 138.78.64.0 через порт маршрутизатора с номером 138.78.64.1.

Узлы в подсетях могут генерировать пакеты для внешней сети. Поскольку внешних сетей может быть большое количество, то чтобы не указывать их все и тем самым сократить размеры таблицы маршрутизации, целесообразно указать всего один маршрут по умолчанию. Маршрут по умолчанию соответствует неизвестному для маршрутизатора M2 адресу сети, поэтому в таблице маршрутизации в полях «номер сети» и «маска» присутствует запись 0.0.0.0. Однако при этом указывается порт маршрутизатора M2, через который следует осуществить передачу, и порт следующего маршрутизатора – куда следует направить пакет. В общем случае записей по умолчанию может быть несколько, тогда маршрутизатор становится перед выбором – по какому маршруту передать пакет. И здесь для выбора оптимального маршрута начинает использоваться дополнительная информация, например скорость канала передачи, надёжность канала, расстояние до сети, выраженное в количестве маршрутизаторов, работоспособен ли данный канал в настоящее время и так далее.

10.4. Маски переменной длины

Деление сети на подсети одинакового размера не всегда обоснованно вследствие того, что большое количество адресов попросту не используется. Для более рачительного использования адресного пространства целесообразно делить сеть на подсети разного размера. Для деления сетей на подсети разных размеров был придуман механизм масок переменной длины.

На рис. 10.10 приведён пример распределения адресного пространства сети компании класса B: 141.56.0.0 на четыре части разного размера. Первая подсеть имеет диапазон адресов в половину всего адресного пространства сети – 2^{15} адресов. Для получения такой подсети необходим один дополнительный разряд адреса (его значение равно 0), поэтому маска для данной подсети будет содержать единицы в 17-ти старших разрядах – 255.255.128.0. Следующая подсеть включает половину от оставшегося адресного пространства – 2^{14} адресов, для получения адреса такой подсети необходим ещё один дополнительный разряд, поэтому маска будет иметь 18 единиц в старших разрядах. При этом подсеть будет иметь адрес 141.56.128.0, а маска – 255.255.192.0 соответственно. Восьмая адресного пространства – 2^{13} адресов исходной сети – была отдана третьей подсети 141.56.192.0, это привело к необходимости использования ещё одного дополнительного разряда адреса, что увеличило маску до 19 разрядов, и она стала иметь значение 255.255.224.0.

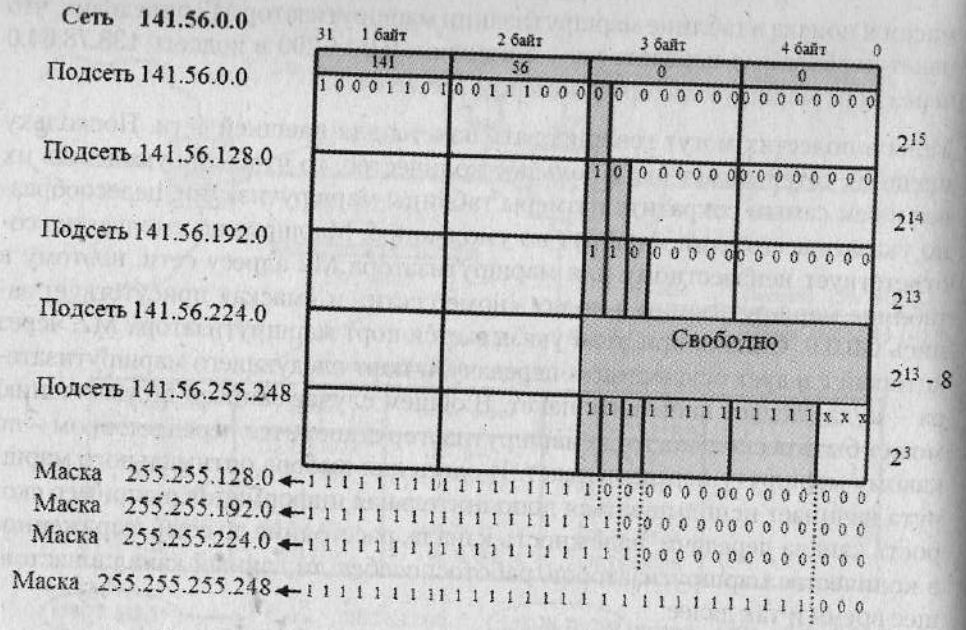


Рис. 10.10. Разделение адресного пространства с помощью масок переменной длины

Для организации связи между маршрутизаторами достаточно иметь всего 4 адреса, но поскольку нам необходим ещё один адрес для компьютера администратора, то для данной подсети определяется диапазон в 8 адресов. Его можно взять из любой части оставшегося свободного адресного пространства. Однако в дальнейшем, скорее всего, может возникнуть необходимость в использовании свободных пока адресов, причём заранее неизвестно, какое количество адресов может потребоваться новым подсетям. Поэтому желательно оставить неиспользуемые адреса единым адресным пространством. В противном случае может возникнуть ситуация, когда для новой подсети не окажется единого свободного адресного пространства достаточного размера. В результате придётся единую подсеть организовывать из нескольких подсетей меньшего размера, что значительно усложнит управление сетью и потребует наличия маршрутизатора с большим количеством портов, что, в конечном счёте, приведёт к увеличению финансовых затрат. В нашем примере для организации подсети, содержащей 8 адресов, были выбраны адреса в конце диапазона. Подсеть получила адрес 141.56.255.248 при маске 255.255.255.248 и при этом отсталось единое свободное адресное пространство размером в 2^{13} - 8 адресов.

Как видно из рис. 10.10, маска переменной длины хоть и позволяет более рационально разделить адресное пространство между подсетями, по срав-

нению с маской постоянной длины, однако размер адресного пространства подсетей кратен степени двойки, поэтому на практике в каждой подсети всегда остаётся некоторое количество не используемых адресов.

На рис. 10.11 приведена структура сети, организованная на основе разделения адресного пространства с помощью масок переменной длины. В таблице маршрутизации маршрутизатора М2 для каждой подсети указана своя маска. Поэтому маршрутизатор при определении адреса соответствующей подсети последовательно просматривает строки таблицы маршрутизации. Из каждой строки маршрутизатор извлекает маску, накладывает её на адрес поступившего пакета и полученный адрес подсети сравнивает со значением указанным в данной строке. Если полученный адрес сети не совпадает с адресом указанным в строке, маршрутизатор переходит на следующую строку, и так до тех пор, пока полученный адрес не совпадёт с указанным адресом в строке. Затем маршрутизатор пересылает пакет на порт, адрес которого указан в этой строке.

Если маршрутизатор не нашёл искомую сеть, то он автоматически дошёл до последней строки, в которой указан маршрут по умолчанию. Маршрутизатор

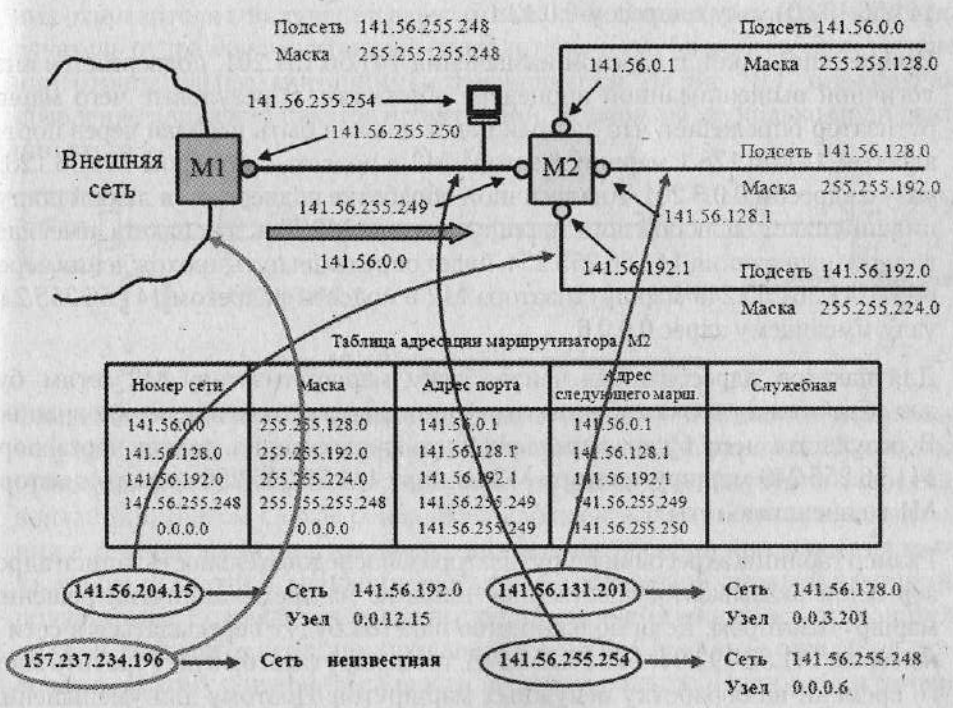


Рис. 10.11. Составная сеть, полученная на основе масок переменной длины

извлекает из строки адреса портов, через которые следует произвести передачу пакета во внешнюю сеть.

Рассмотрим несколько примеров. Пусть пакет, поступивший из внешней сети через маршрутизатор М1, имеет адрес узла 141.56.204.15. Маршрутизатор М2 через свой порт с адресом 141.56.255.249 получил этот пакет. Теперь он начинает последовательный просмотр своей таблицы маршрутизации. Из первой строки он извлекает маску 255.255.128.0, накладывает её на адрес полученного пакета и определяет, что сформированный адрес подсети 141.56.128.0 не соответствует указанному адресу в строке 141.56.0.0. Маршрутизатор переходит на следующую строку, опять извлекает маску, которая теперь имеет значение – 255.255.192.0, накладывает её на адрес пакета и определяет, что сформированный адрес подсети 141.56.192.0 не соответствует указанному адресу подсети 141.56.128.0. Маршрутизатор переходит на следующую строку, извлекает маску 255.255.224.0, накладывает её на адрес пакета и определяет, что сформированный адрес 141.56.192.0 совпадает с указанным в строке адресом. Маршрутизатор нашёл искомую подсеть. Далее он извлекает адрес порта 141.56.192.1, через который необходимо оправить пакет, и направляет пакет в нужную подсеть (адрес подсети 141.56.192.0) узлу с адресом 0.0.12.15.

Пришедший пакет, с адресом назначения 141.56.131.201, подвергается аналогичной вышеописанной процедуре обработки. В результате чего маршрутизатор определяет, что данный пакет должен быть передан через порт с адресом 141.56.128.1 маршрутизатора М2 в подсеть с номером 141.56.128.0 узлу с адресом 0.0.3.201. Аналогичной обработке подвергается любой поступивший пакет на любой порт маршрутизатора М2. Так, для пакета, имеющего адрес назначения 141.56.255.254, будет определён путь прохождения через порт 141.56.255.249 маршрутизатора М2 в подсеть с адресом 141.56.255.248 узлу, имеющему адрес 0.0.0.6.

Для пакетов, адресованных неизвестным маршрутизатору М2 сетям, будет обработана строка таблицы адресации маршрутизатора по умолчанию. В результате чего будет определён путь прохождения пакета через порт 141.56.255.249 маршрутизатора М2 на порт 141.56.255.250 маршрутизатора М1 во внешнюю сеть.

Размер таблицы адресов маршрутизатора и последовательность записи адресов сетей оказывают существенное влияние на время принятия решения маршрутизатором. Если большинство пакетов будут пересылаться в сети с номером 141.56.192.0 и 141.56.255.248, то маршрутизатор будет тратить много времени на обработку ненужных маршрутов. Поэтому для уменьшения времени задержки пакета в маршрутизаторе целесообразно изменить последовательность записей строк в таблице адресов маршрутизатора.

10.5. Обеспечение надёжности передачи данных

Пакеты во время передачи могут искажаться. Для проверки полученных пакетов на целостность применяются контрольные суммы. Однако пакет может вообще не дойти до получателя, в этом случае получатель не сможет правильно интерпретировать полученную информацию (состоящую из нескольких пакетов). Следовательно, кроме контрольной суммы, необходимо иметь ещё методы слежения и обеспечения получения всех пакетов сообщения. Впервые с проблемой утраты пакета (письма, посылки и т.д.) по дороге столкнулись почтовые служащие, поэтому ими и были разработаны методы гарантированной доставки пакетов, основанные на квитанции. Суть методов основывается на отправке отправителю уведомления (квитанции) о получении пакета. Аналогичные, правда, несколько модифицированные, методы применяются и в сетях. В настоящее время наибольшее применение нашли три основных метода: квитирование, групповое квитирование и скользящее окно. Рассмотрим все эти методы более подробно.

Квитирование. Метод квитирования – это самый простой метод уведомления отправителя о получении адресатом пакета. В ответ на каждый пакет получатель отправляет квитанцию отправителю о его благополучном получении, отрицательные квитанции не отправляются. На рис. 10.12 показано направление движения пакетов и квитанций. Возможны несколько модификаций этого метода.

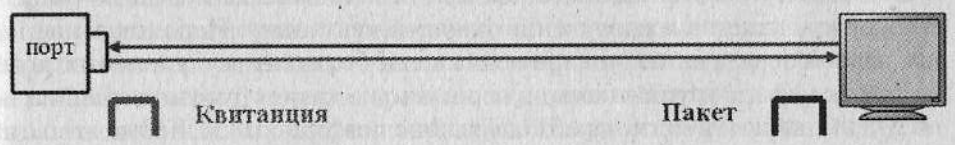


Рис. 10.12. Передача с квитированием

Самый простой способ организации такого метода – это для каждого полученного пакета в адрес отправителя генерировать квитанцию. Причём отправитель может посылать новый пакет только после получения очередной квитанции. В этом случае отправитель должен ждать поступления квитанции в течение времени в несколько раз больше времени прохождения пакета, чтобы исключить ситуации связанные с задержкой оправки квитанции получателем. Если квитанция не поступила за время ожидания, отправитель должен повторить передачу пропавшего пакета. Подобные ожидания приводят к резкому снижению скорости передачи пакетов. Поэтому достаточно часто используется модификация этого метода.

Суть модификации заключается в том, что отправитель не дожидается

получения очередной квитанции от получателя, а посылает пакеты непрерывно. Однако по мере поступления квитанций отправитель должен контролировать ситуацию и отслеживать, на какие пакеты квитанции поступили, а на какие нет, чтобы в дальнейшем осуществить повторную передачу недошедших пакетов. Это, в свою очередь, приводит к усложнению процесса передачи. В любом случае количество пакетов-квитанций, идущих в обратном направлении, равняется количеству переданных пакетов. Огромное количество информации, идущей в обратном направлении, и является основным недостатком данного метода.

Групповое квитирование. Метод группового квитирования направлен на сокращение количества квитанций. Для уменьшения количества квитанций получатель генерирует одну квитанцию не на каждый полученный пакет, а на группу полученных пакетов, как показано на рис. 10.13.

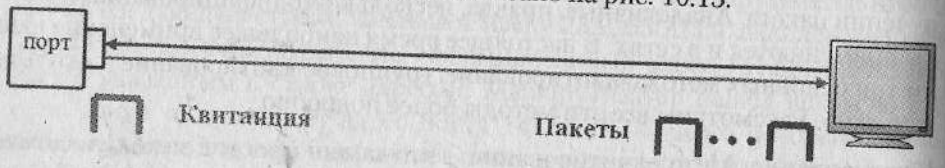


Рис. 10.13. Передача с групповой квитанцией

Передача одной квитанции на группу переданных пакетов позволяет резко уменьшить количество информации, идущей в обратном направлении, пропорционально количеству пакетов в группе. Однако все недостатки, присущие методу квитирования, остаются: невозможность передачи очередной группы пакетов до получения текущей квитанции. Использование непрерывной передачи пакетов приводит к ещё большим трудностям в слежении за тем, какие группы пакетов переданы, на какие группы квитанции не получены, какие группы пакетов переданы повторно и т.д. Кроме этого, не последнюю роль играет время ожидания квитанции. Слишком малое время ожидания может привести к неоправданным повторным передачам, что приведёт к снижению пропускной способности системы. А слишком большое время ожидания может привести к неоправданно большим простоям, связанным с ожиданием несуществующей или «заблудившейся» квитанции.

Метод скользящего окна. В методе скользящего окна разработчики попытались объединить достоинства групповой квитанции с методом непрерывной передачи. Передача информации возможна только в рамках некоторого количества пакетов, которое называется окном (рис. 10.14). Пакеты в окне посылаются непрерывно. Размер окна выбирается из условия, что к моменту поступления квитанции в окне останется не менее 60% непереданных пакетов. Поэтому время ожидания квитанции играет ключевую роль при выборе размера окна, что, в конечном счете, влияет на производительность сети.

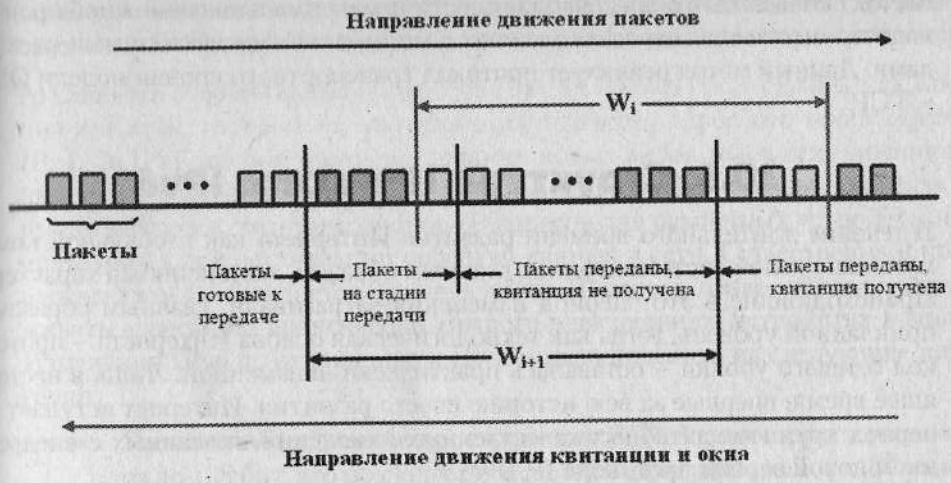


Рис. 10.14. Передача с использованием скользящего окна

При выборе времени ожидания необходимо учитывать скорость и надёжность физических каналов связи и многие другие факторы, например быстродействие и размер буфера маршрутизаторов, размер буфера приёмного устройства, скорость обработки информации приёмником. Поэтому процедура определения времени ожидания достаточно сложна. Она основывается на повторной передаче нескольких контрольных пакетов и получения от получателя квитанции. При этом измеряется время от отправки пакета до получения квитанции – время оборота. Полученные показатели времени усредняются с некоторыми весовыми коэффициентами, возрастающими от замера к замеру. В качестве времени ожидания выбирается средняя величина, увеличенная на коэффициент запаса. На практике коэффициент запаса всегда выбирается больше двух. В сетях с большим разбросом величины времени оборота при определении времени ожидания учитывается и дисперсия этой величины.

На рис. 10.14 показано окно Wi, содержащее некоторое количество пакетов, которые передаются непрерывно. После получения групповой квитанции окно сдвигается в сторону не переданных пакетов на количество пакетов равное размеру группы. В результате пакеты, находящиеся за границей окна справа, – переданы, и на них уже пришли квитанции. Пакеты, находящиеся за границей окна слева, – готовы к передаче, но они не могут быть отправлены, поскольку не находятся в границах окна. Внутри окна существуют два логических пространства: пространство отправленных пакетов, квитанции на которые ещё не получены, и пространство пакетов пока ещё не отправленных.

Метод скользящего окна позволяет достичь максимально возможной прои водительности при передаче пакетов с минимальными накладными расх дами. Данный метод реализует протокол транспортного уровня модели О – TCP.

10.6. Структура IP-пакета IPv6

В течение длительного времени развитие Интернета как глобальной ком муникационной системы носило, главным образом, экстенсивный характер Происходившие в этот период изменения затрагивали, главным образом прикладной уровень, тогда как технологическая основа Интернета – прото кол сетевого уровня – оставалась практически неизменной. Лишь в насто ящее время, впервые за всю историю своего развития, Интернет вступает в период крупномасштабных качественных изменений, связанных с внедре нием новой версии протокола IP, IPv6.

Современный масштаб глобальной IP-сети таков, что проявились те фунда ментальные ограничения, которые были заложены при разработке основ IP протокола 4 десятилетия назад, когда общее количество узлов составляло всего несколько десятков. Например, в 1981 году в сети насчитывалось всего около 200 узлов, в 2000 году – более 75 миллионов, а на конец 2010 года ко личество узлов перевалило за 4 миллиарда.

Одновременно с развитием и усложнением сервисов сети видоизменялась и несущая технология – интернет-протокол. Со временем IP претерпел мно жество модификаций, которые существенно улучшили многие его параме тры. Однако быстрота и перспективы дальнейшего роста сети поставили пе ред IP-протоколом, который используется в настоящее время, ряд «неразре шимых» проблем:

- ограниченное адресное пространство;
- неограниченный рост маршрутных таблиц;
- отсутствие встроенных механизмов обеспечения "качества обслужи вания";
- отсутствие встроенных механизмов автоматической конфигурации узлов;
- отсутствие встроенных средств безопасности;
- неэффективность механизмов поддержки мобильных устройств.

Внедрение протокола IPv6 является одновременно и насущной задачей, и долгосрочной перспективой для сетевых администраторов и операторов сетей общего пользования. С одной стороны, продукты, поддерживающие

IPv6, уже появились на рынке, с другой стороны, доработка и усовершенствование IPv6, вероятно, будут продолжаться и в начавшемся десятилетии.

Активными сторонниками протокола IPv6 являются такие страны, как Япония и Китай, которые не получили достаточного адресного пространства IPv4. За IPv6 ратуют также поставщики новых видов телекоммуникационных услуг. Например, операторы мобильной цифровой телефонной связи – им потребуются сотни миллионов IP-адресов для различных устройств, используемых при организации передачи данных в сети. Существенным преимуществом IPv6 является большая длина IP-адреса, позволяющая поддерживать адресацию практически каждого электронного устройства в мире. Изменения IPv6 по отношению к IPv4 можно поделить на следующие пять групп:

- **Увеличенное адресное пространство**. В IPv6 длина адреса увеличена до 128 бит, что позволяет обеспечить адресацию до $3{,}4 \times 10^{36}$ узлов в сети и упростить автоконфигурацию. Расширены возможности широковещательной рассылки пакетов. Определён новый тип адреса «набор узлов», который используется для посылки запросов группе серверов. Такая адресация предназначена для использования с набором взаимодействующих серверов, чьи адреса не известны клиенту заранее.
- **Фиксированный формат заголовка пакета**. Формат заголовка пакета строго определён и имеет размер в 40 байт. Поэтому ряд полей, аналогичных полям заголовка IPv4, либо исключён, либо перенесён в поле опций, что позволяет уменьшить издержки, связанные с обработкой заголовков пакетов.
- **Улучшенная поддержка расширений и опций**. Введение дополнительных заголовков позволяет облегчить переадресацию пакетов, ослабляет ограничения на длину опций и делает более доступным введение дополнительных опций в будущем.
- **Возможность отмечать потоки данных**. Введена возможность отмечать пакеты, принадлежащие определённым транспортным потокам, для которых отправитель запросил определённую процедуру обработки или обработку данных в реальном масштабе времени.
- **Идентификация и защита частных обменов**. В IPv6 введена спецификация идентификации сетевых объектов или субъектов для обеспечения целостности данных и при желании защиты частной информации.

Весь пакет представляется в виде набора 32-х разрядных слов, как показано на рис. 10.15. Заголовок пакета имеет фиксированный размер в 40 байт.

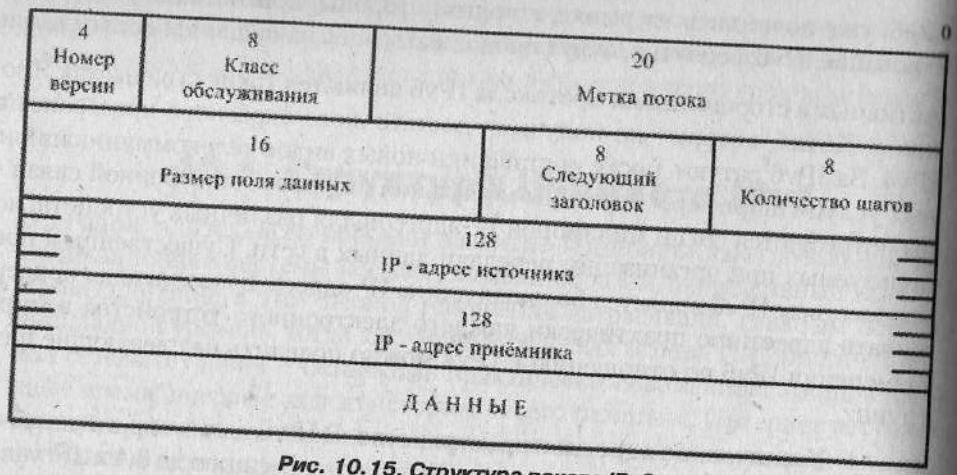


Рис. 10.15. Структура пакета IPv6

Поле «номер версии» содержит 4 бита и определяет номер версии IP-протокола. В данном случае оно должно содержать число 0110, что соответствует шестой версии.

Поле «класс обслуживания» занимает 8 бит и определяет уровень приоритета пакета, поскольку уровень приоритета, в конечном счете, обеспечивает разное время ожидания перед обработкой пакета, то в шестой версии его назвали классом обслуживания. Количество доступных уровней приоритета равно 16, которые разделены на две категории:

- уровни приоритета от 0 до 7 используются для пакетов, которые могут не передаваться при перегруженном канале связи;
- уровни приоритета от 8 до 15 используются для пакетов, которые должны быть обязательно переданы получателю, кроме случая обрыва линии.

Например, приоритет 8 пользователь может назначить пакетам цифрового видео высокой чёткости, пропадание одного из них никоим образом не скажется на качестве полученного видеоизображения. А приоритет 15 пользователь может назначить аудиоданным, поскольку в случае утраты пакета качество фонограммы может быть существенно ухудшено.

Поле «метка потока» занимает 20 разрядов. Это поле может использоваться источником для выделения из общей массы определённой последовательности пакетов (потока). Для каждого из потоков источник может запросить определённый уровень качества обслуживания, отличный от уровня обслуживания, предоставляемого по умолчанию, а также отличный от уровня обслуживания других потоков.

Поле «размер поля данных» занимает 16 разрядов и определяет количество информации, которая следует сразу за 40 байтами заголовка.

Поле «следующий заголовок» занимает 8 бит и определяет тип не обязательного дополнительного заголовка, следующего сразу за заголовком пакета IPv6. Помимо основного заголовка пакет IPv6 может содержать один или несколько дополнительных заголовков, которые используются для предоставления дополнительной информации о пакете. В настоящее время определено 7 типов дополнительных заголовков:

- заголовок опций “шаг-за-шагом” (hop-by-hop);
- заголовок опций получателя 1;
- заголовок маршрутизации;
- заголовок фрагментации;
- заголовок проверки прав доступа (аутентификации);
- заголовок безопасных вложений;
- заголовок опций получателя 2.

Дополнительный заголовок hop-by-hop несёт в себе информацию, которая должна быть рассмотрена и обработана каждым узлом на пути следования пакета, включая отправителя и получателя. Этот заголовок должен быть первым в списке дополнительных заголовков.

Заголовок опций получателя 1 используется для передачи информации по адресу, принадлежащему нескольким разным узлам, которая должна анализироваться одним узлом, находящимся ближе всех от отправителя, в соответствии с метрикой, определённой протоколом маршрутизации.

Заголовок маршрутизации используется отправителем, чтобы заставить пакет посетить один или более промежуточных узлов на пути к месту назначения. Эта функция схожа с опцией принудительной маршрутизации в протоколе IPv4.

Заголовок фрагментации используется отправителем IPv6 для посылки пакетов длиннее, чем принято по умолчанию (1280 байт), до места назначения. (В отличие от IPv4, фрагментация в IPv6 выполняется только узлами-отправителями, а не маршрутизаторами вдоль пути доставки).

Заголовок проверки прав доступа (аутентификации) обеспечивает проверку источника данных и целостность данных, может быть использован самостоятельно. Однако заголовок не обеспечивает шифрование данных.

Заголовок безопасных вложений представляет собой механизм, который

обеспечивает целостность передаваемых данных благодаря шифрованию данных на основе криптографического ключа с использованием асимметричных методов кодирования. Данный механизм безопасности более эффективен, чем ранее использовавшийся в IPv4. Следует отметить, что данный метод используется для решения только определённых задач безопасности и не может использоваться как уникальное средство. Например, этот механизм не может обеспечить конфиденциальности передачи данных от взлома методом анализа передаваемого потока.

Заголовок опций получателя 2 используется для передачи информации только узлу назначения.

Поле «количество шагов» имеет размер 8 бит, оно определяет максимальное количество маршрутизаторов, которое может пройти пакет. После обработки пакета в маршрутизаторе из поля «количество шагов» вычитается единица. Данное поле идентично полю «время жизни» в пакете IPv4.

Поле адреса источника или получателя содержит 128 разрядов, что позволяет иметь $3{,}4 \times 10^{36}$ адресов. Как и в протоколе IPv4, в IPv6 адреса делятся на классы. Класс определяется старшими разрядами адреса. В протоколе IPv6 определено пока всего 3 типа (класса) адресов:

- **Индивидуальный адрес узла**. Пакет, отправленный по индивидуальному адресу, доставляется только одному узлу.
- **Произвольный адрес** – адрес группы узлов сети. Пакет, направленный по произвольному адресу, получает один из узлов группы (ближайший), имеющий наименьшее значение метрики, определяемое протоколом маршрутизации.
- **Групповой** – адрес группы узлов. Пакет, направленный по групповому адресу, должны получить все узлы, имеющие указанный адрес.

Каждый узел должен иметь, по меньшей мере, один индивидуальный адрес. Одному узлу может быть присвоено несколько адресов любого из трёх перечисленных типов.

Индивидуальный адрес или группа индивидуальных адресов могут быть присвоены нескольким узлам. В этом случае группа узлов воспринимается как один логический узел. Это полезно для распределения нагрузки по нескольким физическим узлам.

Старшие разряды адреса определяют структуру оставшейся части адреса: наличие определенных полей, их формат и назначение. Не вдаваясь в детали, заметим лишь, что такая структура адреса позволяет, в частности, реализовать схему иерархической маршрутизации. Эта схема призвана положить предел неограниченному росту таблиц маршрутизации.

Поскольку детальное описание протокола IPv6 и системы адресации принятой в нём выходит за рамки данной книги, то позволим себе ограничиться кратким вышеприведённым описанием.

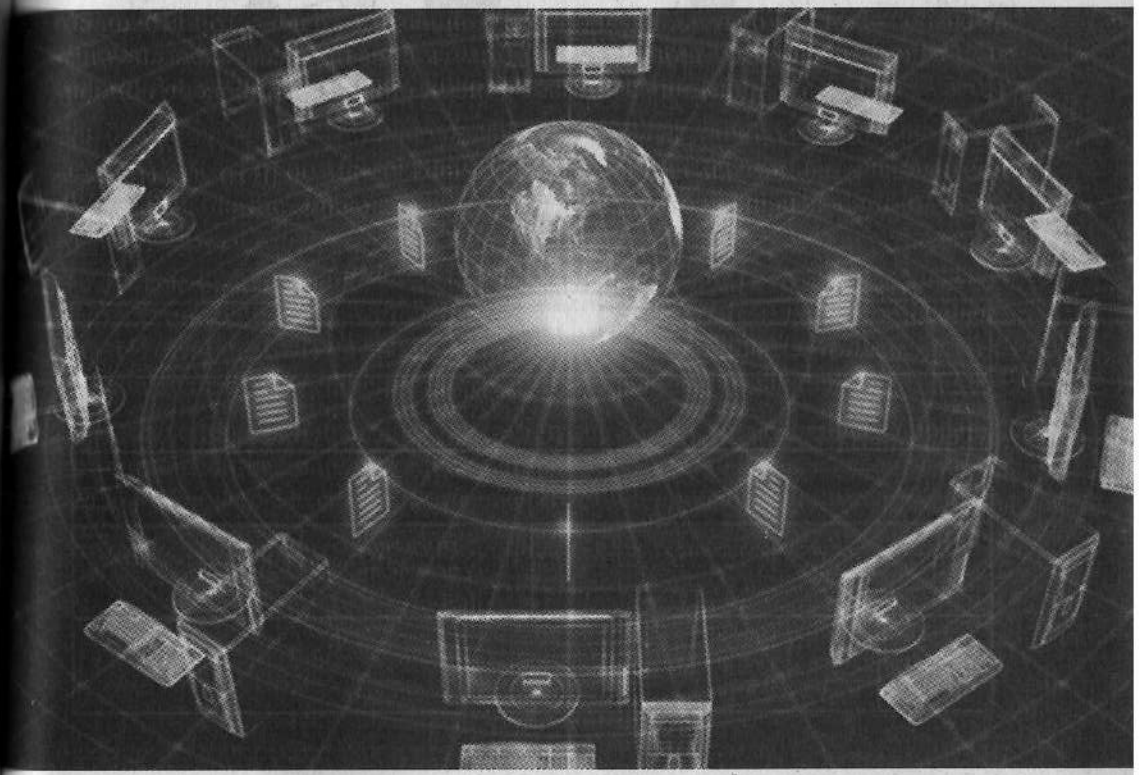
Контрольные вопросы

1. *Почему система адресации коммутаторов не пригодна для сетевого уровня?*
2. *Что собой представляет IP-адрес?*
3. *На какие классы разделено всё пространство IP-адресов?*
4. *Какие существуют специальные IP-адреса?*
5. *Каким образом символьные адреса переводятся в двоичные?*
6. *В чём заключается назначение системы DNS?*
7. *Из каких основных частей состоит IP-пакет?*
8. *Какую структуру имеет заголовок IP-пакета?*
9. *Какие сервисы поддерживаются для IP-пакета?*
10. *В каких случаях используется поле "Смещение" в IP-пакете?*
11. *В чём заключается назначение поля опций?*
12. *Зачем используется механизм масок?*
13. *Каким образом механизм масок постоянной длины преобразует IP-адрес?*
14. *Что позволяет механизм масок переменной длины?*
15. *В чём заключается отличие механизмов масок постоянной длины от масок переменной длины?*
16. *Какие механизмы применяются для обеспечения надёжной передачи информации в сети?*
17. *Какими недостатками обладает метод одиночной квитанции?*
18. *Какими достоинствами обладает метод групповой квитанции?*
19. *Каким образом осуществляется подтверждение принятия пакета в методе скользящего окна?*
20. *Чем обусловлен размер скользящего окна?*
21. *Какими недостатками обладает способ адресация пакетов в протоколе IPv4?*
22. *В чём, на ваш взгляд, заключается необходимость в переходе на протокол IPv6?*

23. Какое количество узлов в сети поддерживает протокол IPv6?
24. Какую структуру имеет заголовок пакета протокола IPv6?
25. Какими достоинствами обладает протокол IPV6, по сравнению с протоколом IPv4?

Глава 11.

МАРШРУТИЗАТОРЫ



Маршрутизатором называется устройство способное выбирать оптимальный маршрут следования пакета в составной сети, имеющей несколько альтернативных путей передачи данных, кроме этого маршрутизатор применяется для деления сети на части, как это было описано в предыдущих главах. На рис. 11.1 приведён пример составной сети, в которой имеется 11 сетей: S1 – S11 и 12 маршрутизаторов.

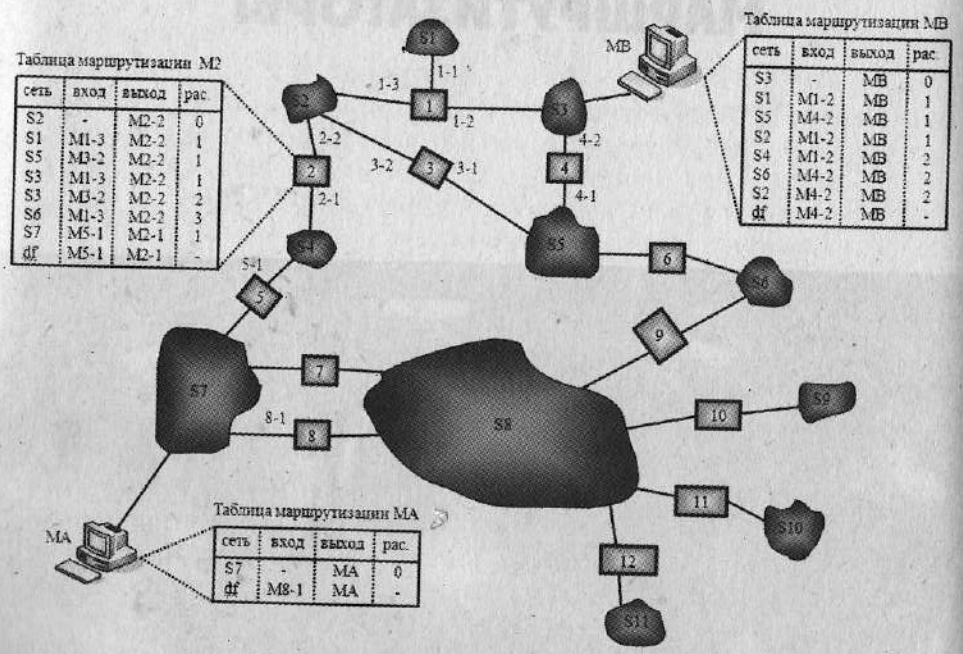


Рис. 11.1. Принципы маршрутизации

Каждый узел и маршрутизатор сети должны иметь таблицу маршрутизации, в которой указаны расстояния до известных ему сетей в маршрутизаторах и ближайшие порты маршрутизаторов, через которые доступны указанные сети. Так, например, узел MB непосредственно подключен к сети S3, поэтому в таблице маршрутизации узла MB имеется запись о том, что сеть S3 непосредственно подключена к узлу MB и расстояние до неё равно нулю. Сеть S1 доступна узлу MB через порт 1-2 первого маршрутизатора, и расстояние до неё равно единице. Для оптимальной передачи информации в нужные сети узел MB должен иметь полную информацию обо всех возможных путях к каждой сети. Например, сеть S2 может быть доступна через порт 1-2 первого маршрутизатора при расстоянии 1 или через порт 4-2 четвёртого маршрутизатора с расстоянием 2.

При беглом взгляде на рис. 11.1 понятно, что для указания всех альтернативных путей таблица маршрутизации должна содержать десятки строк, что приведёт к дополнительным задержкам при выборе маршрута, поскольку придётся каждый раз просматривать всю таблицу маршрутизации с начала и до конца. С целью уменьшения размеров таблицы маршрутизации администраторы указывают обычно маршруты только до ближайших сетей, находящихся на расстоянии в один маршрутизатор. А в том случае, когда запись об искомой сети отсутствует, указывается путь по умолчанию через порт заранее определённого маршрутизатора. В нашем примере для узла MB по умолчанию все пакеты будут направляться на порт 4-2 четвёртого маршрутизатора. Для ещё большего сокращения размеров таблицы маршрутизации администратор может указать всего два пути: один до ближайшей сети (к которой подключен узел) и второй по умолчанию. В качестве такого примера приведена таблица маршрутизации узла MA. Аналогичную таблицу маршрутизации должны иметь все маршрутизаторы. На рис. 11.1, для примера, приведена таблица маршрутизации маршрутизатора 2.

Таблицы маршрутизации может заполнять сам администратор, однако при больших размерах составной сети, в которой к тому же появляются новые сети и маршрутизаторы, поддержание таблиц маршрутизации на маршрутизаторах крайне затруднительно, поскольку администратор должен оперативно получать информацию от других администраторов о появлении новых сетей и маршрутизаторов. Для облегчения жизни администраторов целесообразно организовать процесс автоматической рассылки информации об изменениях топологии составной сети, то есть организовать процесс таким образом, чтобы маршрутизаторы сами обменивались информацией о возникших изменениях и самостоятельно корректировали свои таблицы маршрутизации.

Прежде чем приступить к рассмотрению основ и принципов маршрутизации, следует остановиться на определении двух основных понятий: **маршрутизируемый** и **маршрутизирующий** протоколы, которые будут в дальнейшем использоваться. К сожалению, эти два термина имеют очень схожее произношение, особенно в английском написании: **routed** и **routing**, что часто приводит к путанице.

Маршрутизируемый протокол – это сетевой протокол, которому достаточно только сетевого адреса, чтобы доставить пакет по назначению на основе используемой схемы адресации. Примерами маршрутизируемых протоколов являются: IP, IPX, AppleTalk, DECnet.

Маршрутизирующий протокол – это протокол, который предоставляет маршрутизаторам механизмы обмена маршрутной информацией друг с другом для обновления записей и поддержки таблиц маршрутизации. Примерами

протоколов маршрутизации стека TCP/IP являются:

- **RIP** – протокол маршрутной информации;
- **EIGRP** – усовершенствованный протокол маршрутизации внутреннего шлюза;
- **OSPF** – открытый протокол предпочтения кратчайшего пути.

Маршрутизация – это процесс, при котором осуществляется передача пакетов от отправителя к получателю при помощи протокола маршрутизации.

Все маршрутизаторы должны иметь свои (локальные) таблицы маршрутизации. Они используются маршрутизатором для определения наилучшего пути от источника к пункту назначения. Таблица маршрутизации содержит следующие записи:

- механизм, по которому был получен маршрут;
- адрес получателя в виде сети или подсети;
- административное расстояние;
- метрика маршрута;
- адрес порта маршрутизатора, через который доступна сеть-получатель, расположенного на расстоянии одной пересылки;
- время создания записи в таблице;
- выходной порт маршрутизатора, через который доступна сеть-получатель.

Маршрутизатор производит выбор оптимального маршрута к сетям получателям. Поскольку одновременно на маршрутизаторе может быть запущено сразу несколько протоколов маршрутизации, то возникает проблема выбора между маршрутами, полученными от разных протоколов маршрутизации. Для выбора оптимального маршрута из маршрутов, полученных от разных протоколов маршрутизации, маршрутизаторы используют концепцию административного расстояния. Административное расстояние рассматривается как мера достоверности источника информации о маршруте. В таблице 11.1 представлены административные расстояния, применяемые в маршрутизаторах Cisco для различных протоколов маршрутизации.

Таблица 11.1. *Административные расстояния, принятые в маршрутизаторах Cisco*

Источник информации о маршруте	Административное расстояние
Прямое соединение	0
Статический маршрут	1

Суммарный маршрут EIGRP	5
Внешний BGP	20
Внутренний EIGRP	90
IGRP	100
OSPF	110
IS–IS	115
RIPv1 RIPv2	120
Внешний EIGRP	170
Внутренний BGP	200
Неизвестный	255

Определение оптимального маршрута является особенностью присущей любому протоколу. Каждый протокол имеет свою меру того, что является лучшим (оптимальным). Маршрутизаторы характеризуют маршрут к сети с помощью некоторой интегральной величины называемой метрикой маршрута, при этом чем меньше значение метрики, тем оптимальнее маршрут. В алгоритмах маршрутизации для вычисления метрики наиболее часто используются следующие параметры:

- ширина полосы пропускания канала связи представляет собой оценку объёма информации, которая может быть передана;
- задержка – представляет собой время перемещения пакета по каналу связи от отправителя к получателю. Она зависит от пропускной способности промежуточных каналов связи, размера очередей в портах маршрутизаторов, загрузки сети и физического расстояния;
- загрузка – средняя загруженность канала связи в единицу времени;
- надёжность – относительное количество ошибок на канале связи;
- количество переходов – количество маршрутизаторов, которые должен пройти пакет на своём пути до получателя;
- стоимость – определяет цену передачи пакета в денежных единицах или других единицах измерения, назначаемых администратором.

К пункту назначения может существовать множество путей, но протокол маршрутизации должен выбрать один путь как наилучший и поместить его в таблицу маршрутизации. Однако многие протоколы маршрутизации поддерживают механизм балансировки нагрузки, при котором в таблицу маршрутизации могут быть записаны несколько возможных маршрутов к узлу-получателю, и передача пакетов будет осуществляться по каждому из маршрутов.

Для построения таблиц маршрутизации могут использоваться следующие методы:

- прямое соединение;
- статическое соединение (маршрутизация);
- соединение (маршрутизация) по умолчанию;
- динамическое соединение (маршрутизация).

Прямое соединение определяет непосредственное соединение порта маршрутизатора с сетью, оно всегда является наилучшим способом маршрутизации, поскольку маршрутизатору известна непосредственно присоединённая к нему сеть и пакеты, предназначенные для неё, сразу направляются на соответствующий порт маршрутизатора без использования протоколов маршрутизации.

Статические маршруты – это маршруты к сетям, которые администратор вручную вносит в таблицу маршрутизации, указав адрес порта следующего соседнего маршрутизатора. Если указанный в маршруте порт маршрутизатора становится недоступными, то и маршрут к сети получателю становится недоступным. Статическая маршрутизация может быть использована в ситуациях когда:

- администратор нуждается в полном контроле маршрутов;
- необходимо резервирование динамических маршрутов;
- есть сети достижимые единственно возможным путём;
- нежелательно иметь служебный трафик для обновления таблиц маршрутизации;
- используются устаревшие маршрутизаторы, не имеющие необходимого уровня вычислительных возможностей для поддержки динамических протоколов маршрутизации.

Наиболее предпочтительной топологией для использования статической маршрутизации является звездообразная топология. В этом случае маршрутизаторы, подключенные к центральной точке сети, имеют только один маршрут для всех пакетов, которые будут проходить через центральный узел сети. И маршрутизатор (или маршрутизаторы) в центральной части сети имеют статические маршруты до всех удалённых узлов.

Однако со временем такая сеть может вырасти до десятков и сотен маршрутизаторов с произвольным количеством подключённых к ним подсетей. Количество статических маршрутов в таблицах маршрутизации будет увеличиваться пропорционально увеличению количества маршрутизаторов в сети. Каждый раз при добавлении новой подсети или маршрутизатора администратор должен будет добавлять новые маршруты в таблицы маршрутизации на

всех необходимых маршрутизаторах. При таком подходе может наступить момент, когда большую часть своего рабочего времени администратор будет заниматься поддержкой таблиц маршрутизации в сети.

Другой недостаток статической маршрутизации проявляется при изменении топологии корпоративной сети. При этом администратор должен вручную вносить все изменения в таблицы маршрутизации маршрутизаторов, на которые повлияли изменения в топологии сети.

Бывают ситуации, когда маршрутизатору не нужно знать обо всех сетях в топологии (рис. 11.1), тогда он может посылать все пакеты с неизвестным ему адресом назначения по специальному маршруту – маршруту по умолчанию.

При использовании протоколов динамической маршрутизации администратор сети конфигурирует выбранный протокол на каждом маршрутизаторе в сети. После этого маршрутизаторы начинают обмен информацией об известных им сетях и их состояниях. Причём маршрутизаторы обмениваются информацией только с теми маршрутизаторами, где запущен тот же протокол динамической маршрутизации. Когда происходит изменение топологии сети, информация об этих изменениях автоматически распространяется по всем маршрутизаторам, и каждый маршрутизатор вносит необходимые изменения в свою таблицу маршрутизации.

Успешное функционирование динамической маршрутизации зависит от выполнения маршрутизатором двух основных функций: поддержки таблицы маршрутизации в актуальном состоянии и своевременного распространения информации об известных ему сетях. При распространении информации о сетях механизм динамической маршрутизации использует один из протоколов маршрутизации, который определяет набор правил, используемых для связи с соседними маршрутизаторами. Обычно задаются следующие параметры: каким образом распространяются обновления маршрутов; какая информация содержится в обновлениях; как часто рассылаются обновления; каким образом выполняется поиск получателей обновлений.

В технологии маршрутизации существует два понятия – автономная система и домен маршрутизации (рис. 11.2).

Автономная система (autonomous system, AS) – это набор сетей, которые находятся под единым административным управлением и в которых используются единая стратегия и правила маршрутизации. Автономная система для внешних сетей представляется как некий единый объект.

Домен маршрутизации – это совокупность сетей и маршрутизаторов, использующих один и тот же протокол маршрутизации. На рис. 11.2 представлены три домена маршрутизации, объединённые в одну автономную систему.

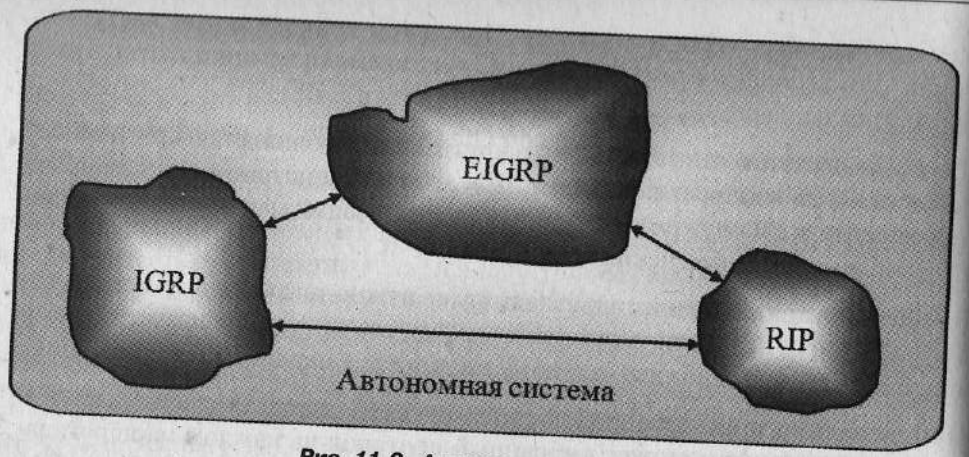


Рис. 11.2. Автономная система

Сеть Internet изначально строилась как сеть, объединяющая большое количество существующих систем. Поэтому в её структуре была выделена одна магистральная сеть, к которой присоединялись автономные системы.

В сети Internet термин автономная система используется для описания крупных логически объединённых сетей, например сетей провайдеров Internet. Каждая AS имеет уникальный 16-ти разрядный двоичный идентификатор (номер). Номера автономным системам выдаёт и регистрирует Американский реестр интернет-номеров (American Registry of Internet Numbers – ARIN), при этом для публичных сетей отведён диапазон номеров в пределах от 0 до 64511. Диапазон номеров в пределах 64512 – 65534 отведён частным автономным системам, а автономная система с номером 65535 зарезервирована под магистраль, которая призвана соединять между собой все автономные системы.

Для маршрутизации пакетов внутри автономной системы используются внутренние протоколы, которые имеют общее название IGP (протоколы внутреннего шлюза). К ним относится любой протокол маршрутизации, используемый исключительно внутри автономной системы, например RIP, EIGRP и OSPF. Каждый внутренний протокол представляет один домен маршрутизации внутри AS (рис. 11.3). Для передачи информации между доменами внутри автономной системы используются граничные или междоменные маршрутизаторы.

Внешние протоколы – EGP (протоколы внешнего шлюза) обеспечивают маршрутизацию между автономными системами и магистральной сетью для транзита пакетов в другие AS (рис. 11.4). Протокол BGP (протокол пограничного шлюза) является одним из наиболее известных межсистемных протоколов маршрутизации.

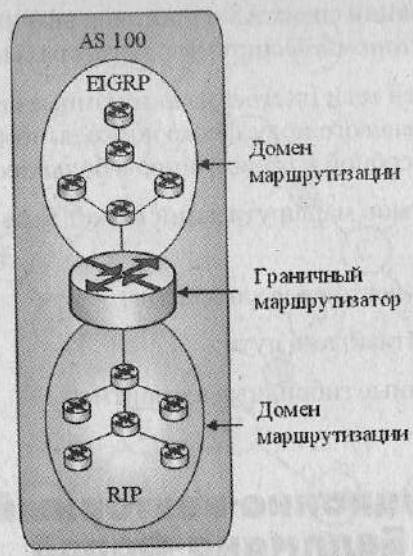


Рис. 11.3. Внутренние протоколы маршрутизации

Внешние протоколы маршрутизации только распознают автономные системы, игнорируя внутренние протоколы маршрутизации. Граничные маршрутизаторы автономных систем поддерживают какой-либо тип внутреннего

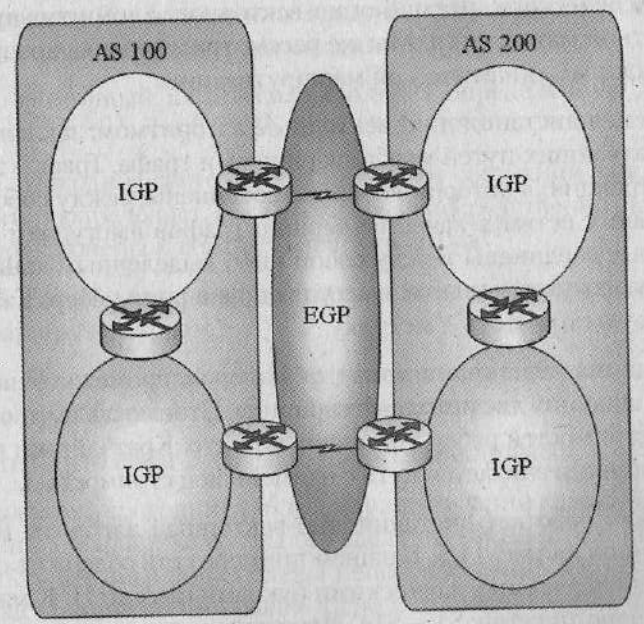


Рис. 11.4. Внешние протоколы маршрутизации

протокола маршрутизации своих AS и протокол внешнего шлюза для соединения собственной автономной системы с магистральной сетью.

Смысл разделения всей сети Internet на автономные системы заключается в реализации многоуровневого модульного подхода, необходимого для любой крупной системы, способной к расширению в больших масштабах.

Большинство алгоритмов маршрутизации может быть отнесено к одной из трех категорий:

- дистанционно-векторные алгоритмы;
- алгоритмы кратчайшего пути;
- сбалансированные гибридные алгоритмы.

11.1. Дистанционно-векторный алгоритм Беллмана-Форда

Все дистанционно-векторные протоколы маршрутизации основываются на дистанционно-векторном алгоритме, который был впервые описан Фордом и Фулкерсоном. Их работа основана на математическом аппарате, предложенным Беллманом.

Изучение классического дистанционно-векторного алгоритма проводится в курсе дискретной математики. Мы же рассмотрим версию, которая используется в протоколах динамической маршрутизации.

Задача, решаемая дистанционно-векторным алгоритмом, заключается в нахождении кратчайших путей между вершинами графа. Граф – это математическая абстракция, в которой вершины соединены между собой ребрами. Применительно к сетям в качестве вершин графов выступают маршрутизаторы, которые соединены между собой либо выделенным каналом связи, либо через промежуточные сети, выступающие в роли рёбер. Каждое ребро имеет некоторую стоимость – метрику.

Путь между двумя вершинами является набором промежуточных рёбер и вершин, соединяющих две исходные вершины. Стоимость пути определяется как сумма стоимостей ребер, составляющих его. Кратчайшим путём между двумя вершинами считается путь с наименьшей стоимостью.

Рассмотрим, как работает дистанционно-векторный алгоритм на примере сети, приведённой на рис. 11.5. В нашем примере сеть содержит семь маршрутизаторов, обозначенных латинскими буквами от A до H. К маршрутизаторам подключено 14 сетей: S1 – S14. Некоторые сети являются конечными

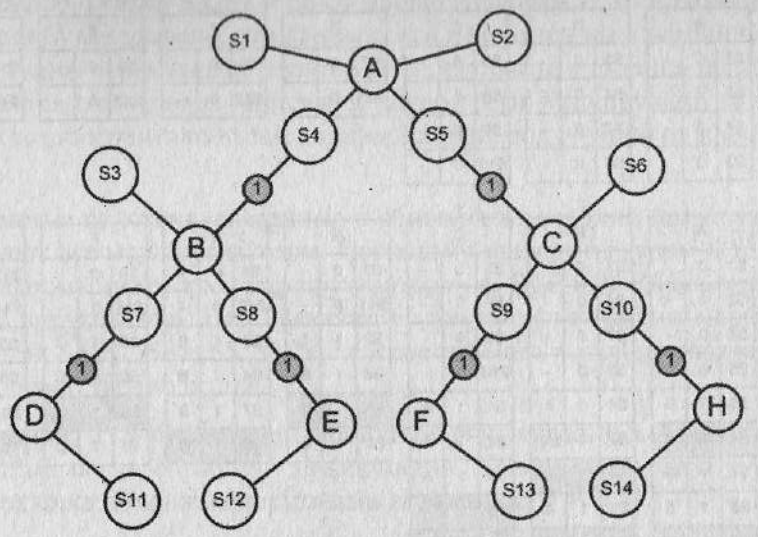


Рис. 11.5. Пример топологии сети

(то есть за ними никаких сетей нет), а через остальные осуществляется связь с маршрутизаторами. Все соединения между маршрутизаторами для простоты имеют одинаковые метрики, равные единице.

Дистанционно-векторный алгоритм выполняет определённую последовательность шагов.

- **Шаг 1.** Исходное состояние. В исходном состоянии каждый маршрутизатор определяет, какие сети его непосредственно окружают, то есть, подключены к портам маршрутизатора.
- **Шаг 2.** Рассылка маршрутизаторами таблиц маршрутизации соседним маршрутизаторам.
- **Шаг 3.** Получение таблиц маршрутизации от соседних маршрутизаторов и обработка полученной информации.

Следует заметить, что шаги 2 и 3 выполняются циклически.

На рис.11.6. показано состояние таблиц маршрутизации каждого маршрутизатора. На Шаге 1 маршрутизаторы определяют, какие сети к ним подключены. Поскольку все сети подключены непосредственно к портам маршрутизатора, то расстояние до них равно нулю и промежуточные маршрутизаторы отсутствуют.

Шаг 1

A		
S1	0	-
S2	0	-
S4	0	-
S5	0	-

B		
S3	0	-
S4	0	-
S7	0	-
S8	0	-

C		
S5	0	-
S6	0	-
S9	0	-
S10	0	-

D		
S7	0	-
S11	0	-

E		
S8	0	-
S12	0	-

F		
S9	0	-
S13	0	-

H		
S10	0	-
S14	0	-

Шаг 2

A		
S1	0	-
S2	0	-
S4	0	-
S5	0	-
S3	1	B
S4	1	B
S7	1	B
S8	1	B
S5	1	C
S6	1	C
S9	1	C
S10	1	C

B		
S3	0	-
S4	0	-
S7	0	-
S8	0	-
S1	1	A
S2	1	A
S4	1	A
S5	1	A
S7	1	D
S11	1	D
S8	1	E
S12	1	E

C		
S5	0	-
S6	0	-
S9	0	-
S10	0	-
S1	1	A
S2	1	A
S4	1	A
S5	1	A
S9	1	F
S13	1	F
S10	1	H
S14	1	H

D		
S7	0	-
S11	0	-
S3	1	B
S4	1	B
S7	1	B
S8	1	B

E		
S8	0	-
S12	0	-
S3	1	B
S4	1	B
S7	1	B
S8	1	B

F		
S9	0	-
S13	0	-
S5	1	C
S6	1	C
S9	1	C
10	1	C

H		
S10	0	-
S14	0	-
S5	1	C
S6	1	C
S9	1	C
10	1	C

Рис. 11.6. Построение таблиц маршрутизации

На следующем шаге (Шаг 2) все маршрутизаторы передают свои таблицы маршрутизации ближайшим маршрутизаторам. В нашем примере (рис. 11.5) у маршрутизатора А ближайшими соседями являются маршрутизаторы В и С, которые и передают маршрутизатору А свои таблицы маршрутизации. Переданные таблицы помещаются в таблицу маршрутизации маршрутизатора А. В нашем примере все данные таблицы маршрутизации маршрутизатора В имеют синий цвет, а маршрутизатора С – зелёный. Далее маршрутизатор А просматривает свою таблицу маршрутизации на предмет обнаружения новой информации об известных ему ранее сетях. И обнаруживает, что информация, полученная о сети S4 от маршрутизатора В, говорит о том, что эта сеть находится на расстоянии одной метрики, да ещё и за маршрутизатором В. Поскольку метрика в полученной информации хуже, чем в имеющейся (метрика равна нулю), то маршрутизатор удаляет эту строку из таблицы маршрутизации (строка отмечена коричневым цветом). Та же участь постигает и информацию о сети S5, полученную от маршрутизатора С.

Новая информация остаётся в таблице маршрутизации и означает, через какой маршрутизатор доступны новые сети. Поскольку маршрутизатор А получил

информацию о сетях S3, S7 и S8 от маршрутизатора В, то в таблице маршрутизации будет указано, что данные сети находятся на расстоянии единица и доступны через маршрутизатор В. В результате в таблице маршрутизации остаётся только одна запись для каждой сети. Маршрутизатор А поступает так же, как описано выше, с информацией, полученной от маршрутизатора С.

Аналогичные действия, связанные с обновлением таблиц маршрутизации, выполняют все маршрутизаторы. Поскольку маршрутизаторы В и С получили информацию от трёх маршрутизаторов А, D, E и A, F, H соответственно, то маршрутизаторы В и С удаляют по три записи из таблиц маршрутизации о сетях S4, S7, S8 и S5, S9, S10 соответственно, а маршрутизаторы D, E, F и H соответственно по две.

Шаг 2 на рис. 11.7 отображает таблицы маршрутизации маршрутизаторов, полученные после обработки информации. Далее маршрутизаторы опять начинают цикл обмена маршрутными таблицами.

После очередного обмена таблицами маршрутизации и обработки полученной информации (Шаг 3) таблицы маршрутизации маршрутизаторов будут иметь вид, показанный на рис. 11.7. Здесь следует отметить, что после шага 2 каждый маршрутизатор имел информацию о сетях, которые располагались на расстоянии с метрикой 1. Однако вновь пришедшая информация указывает на то, что некоторые сети находятся на расстоянии единица (метрика равна единице) от маршрутизатора, пославшего информацию. Поэтому маршрутизатор, принявший подобную информацию, увеличивает метрику к таким сетям на единицу и указывает, через какой ближайший маршрутизатор они доступны. Для маршрутизатора А все сети будут доступны только через два ближайших маршрутизатора В и С, следовательно в таблице маршрутизации указываются только эти два маршрутизатора.

Далее маршрутизаторы опять начинают цикл обмена таблицами маршрутизации. После обработки полученной информации (Шаг 4) таблицы маршрутизации маршрутизаторов будут иметь вид, как показано на рис. 11.7.

Как видно из рисунка 11.7, таблицы маршрутизации маршрутизаторов D, E, H и F не содержат информации о двух сетях из четырёх S11, S12, S13 и S14. Поэтому необходим ещё один цикл обмена информацией, после чего все сети будут достижимы из любой сети посредством некоторого рационального маршрута.

Очевидно, что после построения таблиц маршрутизации, если все маршрутизаторы и каналы связи постоянно работоспособны, обмен маршрутными таблицами можно производить достаточно редко. Однако в реальных сетях постоянно происходят изменения – нарушается как работоспособность

Шаг 2

A		
S1	0	-
S2	0	-
S4	0	-
S5	0	-
S3	1	B
S7	1	B
S8	1	B
S6	1	C
S9	1	C
S10	1	C

B		
S3	0	-
S4	0	-
S7	0	-
S8	0	-
S1	1	A
S2	1	A
S5	1	A
S11	1	D
S12	1	E

C		
S5	0	-
S6	0	-
S9	0	-
S10	0	-
S1	1	A
S2	1	A
S4	1	A
S13	1	F
S14	1	H

D		
S7	0	-
S11	0	-
S3	1	B
S4	1	B
S8	1	B

E		
S8	0	-
S12	0	-
S3	1	B
S4	1	B
S7	1	B

F		
S9	0	-
S13	0	-
S5	1	C
S6	1	C
10	1	C

H		
S10	0	-
S14	0	-
S5	1	C
S6	1	C
S9	1	C

Шаг 3

A		
S1	0	-
S2	0	-
S4	0	-
S5	0	-
S3	1	B
S7	1	B
S8	1	B
S6	1	C
S9	1	C
S10	1	C
S11	2	B
S12	2	B
S13	2	C
S14	2	C

B		
S3	0	-
S4	0	-
S7	0	-
S8	0	-
S1	1	A
S2	1	A
S5	1	A
S11	1	D
S12	1	E
S6	2	A
S9	2	A
S10	2	A

C		
S5	0	-
S6	0	-
S9	0	-
S10	0	-
S1	1	A
S2	1	A
S4	1	A
S13	1	F
S14	1	H
S3	2	A
S7	2	A
S8	2	A

D		
S7	0	-
S11	0	-
S3	1	B
S4	1	B
S8	1	B
S1	2	B
S2	2	B
S5	2	B
S12	2	B

E		
S8	0	-
S12	0	-
S3	1	B
S4	1	B
S7	1	B
S1	2	B
S2	2	B
S5	2	B
S11	2	B

F		
S9	0	-
S13	0	-
S5	1	C
S6	1	C
10	1	C
S1	2	C
S2	2	C
S4	2	C
S14	2	C

H		
S10	0	-
S14	0	-
S5	1	C
S6	1	C
S9	1	C
S1	2	C
S2	2	C
S4	2	C
S13	2	C

Шаг 4

A		
S1	0	-
S2	0	-
S4	0	-
S5	0	-
S3	1	B
S7	1	B
S8	1	B
S6	1	C
S9	1	C
S10	1	C
S11	2	B
S12	2	B
S13	2	B
S14	2	B

B		
S3	0	-
S4	0	-
S7	0	-
S8	0	-
S1	1	A
S2	1	A
S5	1	A
S11	1	D
S12	1	E
S6	2	A
S9	2	A
S10	2	A
S13	3	A
S14	3	A

C		
S5	0	-
S6	0	-
S9	0	-
S10	0	-
S1	1	A
S2	1	A
S4	1	A
S13	1	F
S14	1	H
S3	2	A
S7	2	A
S8	2	A
S11	3	A
S12	3	A

D		
S7	0	-
S11	0	-
S3	1	B
S4	1	B
S8	1	B
S1	2	B
S2	2	B
S5	2	B
S12	2	B
S6	3	B
S9	3	B
S10	3	B

E		
S8	0	-
S12	0	-
S3	1	B
S4	1	B
S7	1	B
S1	2	B
S2	2	B
S5	2	B
S11	2	B
S6	3	B
S9	3	B
S10	3	B

F		
S9	0	-
S13	0	-
S5	1	C
S6	1	C
10	1	C
S1	2	C
S2	2	C
S4	2	C
S14	2	C
S3	3	C
S7	3	C
S8	3	C

H		
S10	0	-
S14	0	-
S5	1	C
S6	1	C
S9	1	C
S1	2	C
S2	2	C
S4	2	C
S13	2	C
S3	3	C
S7	3	C
S8	3	C

Рис. 11.7. Таблицы маршрутизации

маршрутизаторов, так и каналов связи, изменяются топологии сети – добавляются или наоборот выводятся как сами маршрутизаторы, так и каналы связи. Поэтому необходимо постоянно осуществлять обмен копиями таблиц маршрутной информации между маршрутизаторами.

Маршрутизацию на основе дистанционно-векторного алгоритма иногда называют «маршрутизацией по слухам», поскольку маршрутизаторы не знают ничего о маршрутизаторах на пути к известным им сетям.

Дистанционно-векторный алгоритм реализован в протоколе RIP, причём для IP-сетей существуют две версии. Первая версия работает только с адресами стандартных классов – A, B или C. Вторая же версия – поддерживает маски сетей. Поскольку построение таблиц маршрутизации обеими версиями протокола RIP принципиально не отличается, то в дальнейшем при указании на протокол RIP будет иметься в виду первая версия.

Протокол RIP хорошо справляется с ситуацией, когда изменение сети происходит в «большую» сторону, то есть к существующей сети добавляются новые сети или новые маршрутизаторы, этому способствуют регулярные рассылки таблиц маршрутизации. Действительно, при изменении сети рассылки ближайшим соседям приведут к тому, что постепенно эта информация распространится по сети и станет известна всем маршрутизаторам.

При изменении сети в меньшую сторону – связанном с потерей маршрута (это может произойти в результате выхода канала связи или маршрутизатора из строя) – протокол RIP не имеет возможности сообщить соседям о таком изменении, поскольку в составе сообщений отсутствует поле, указывающее на то, что путь к данной сети больше не существует. Это приводит к появлению петли маршрутизации. В доказательство сказанного рассмотрим пример сети, изображённой на рис. 11.8.

Непосредственно перед выходом из строя сети S1 все маршрутизаторы имеют согласованные таблицы маршрутизации. Предположим, что для маршрутизатора R3 наилучший маршрут к сети S1 пролегает через маршрутизатор R2 и в своей таблице маршрутизации маршрутизатор R3 имеет запись о расстоянии до сети S1, равном 3 переходам.

После выхода сети S1 из строя маршрутизатор R5 пересылает маршрутизатору R1 обновлённую таблицу маршрутизации, содержащую эту информацию. После получения обновления маршрутизатор R1 прекращает направлять пакеты в сеть S1, однако маршрутизаторы R2, R3 и R4 продолжают это делать, поскольку ещё не получили информацию о сбое в сети S1. Через некоторое время маршрутизатор R1 отправляет своё обновление маршрутной таблицы маршрутизаторам R2 и R4, в результате чего они прекращают направлять пакеты в сеть S1, но маршрутизатор R3, ещё не получивший

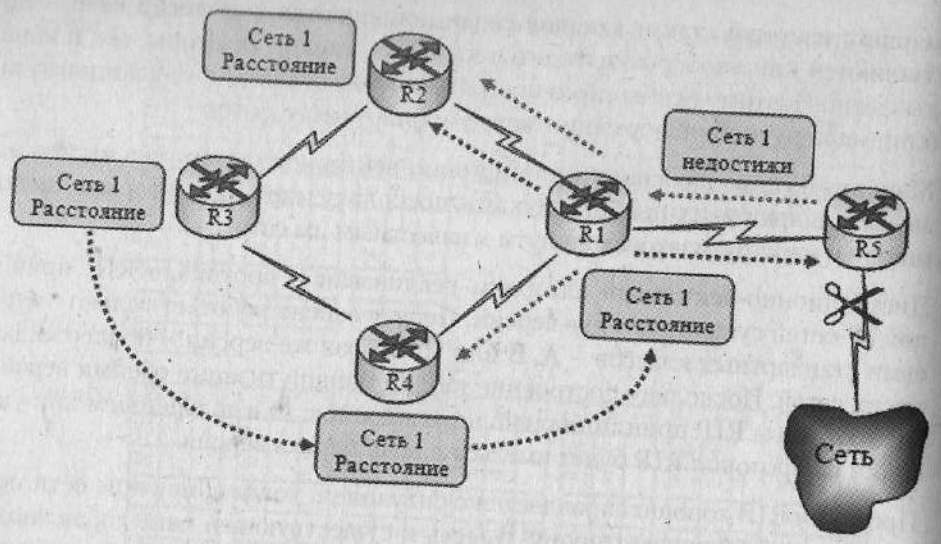


Рис. 11.8. Образование петли маршрутизации

обновление маршрутной информации, по-прежнему посылает пакеты в сеть S1 через маршрутизатор R2.

По алгоритму работы дистанционно-векторных протоколов маршрутизации все маршрутизаторы регулярно, то есть через определённые промежутки времени, должны рассылать своим соседям маршрутную информацию. Поэтому маршрутизатор R3 посылает свою таблицу маршрутизации маршрутизатору R4, указывая, что он имеет маршрут до сети S1 через маршрутизатор R2. Маршрутизатор R4 изменяет свою таблицу, отражая эту хорошую, но неправильную информацию, и передаёт эти сведения дальше маршрутизатору R1 (увеличив расстояние на единицу). Маршрутизатор R1 распространяет её маршрутизаторам R2 и R5. Теперь любой пакет, имеющий назначением сеть S1, движется по кольцевому маршруту (петле) от маршрутизатора R3 к маршрутизатору R2, далее к R1 и R4 и вновь к маршрутизатору R3.

Кроме этого, пакеты обновления с информацией о сети S1 будут продолжать ходить по кругу до тех пор, пока какой-нибудь другой процесс не сможет остановить это зацикливание. Для разрешения подобных ситуаций необходимо применять специальные методы уведомления о недействительности маршрута. Существует несколько методов: счёт до бесконечности, расщепление горизонта, обратное обновление, таймеры удержания и мгновенное обновление.

Метод счёта до бесконечности

Каждый раз, когда пакет обновления попадает в маршрутизатор, маршрутизатор увеличивает значение метрики (количество переходов) до соответствующих новых сетей на единицу. Если не применять никаких ограничивающих мер, то счёт будет производиться до переполнения разрядной сетки, а это очень большие величины, и потребует большого количества времени, поэтому такая ситуация получила название «счёт до бесконечности». В протоколе RIP величину бесконечности приняли равной 15. Как только значение метрики превысит максимальную величину, сеть S1 будет считаться недостижимой. При таком подходе протокол маршрутизации позволит существовать маршрутизации по кругу до тех пор, пока метрика не превысит максимально допустимое значение, то есть некоторое заранее определённое время.

Метод расщепления горизонта

Другим возможным источником маршрутизации по кругу является ситуация, когда неправильная информация, посылаемая назад маршрутизатору, противоречит информации, посылаемой им самим. Рис. 11.9 поясняет, как возникает такая проблема.

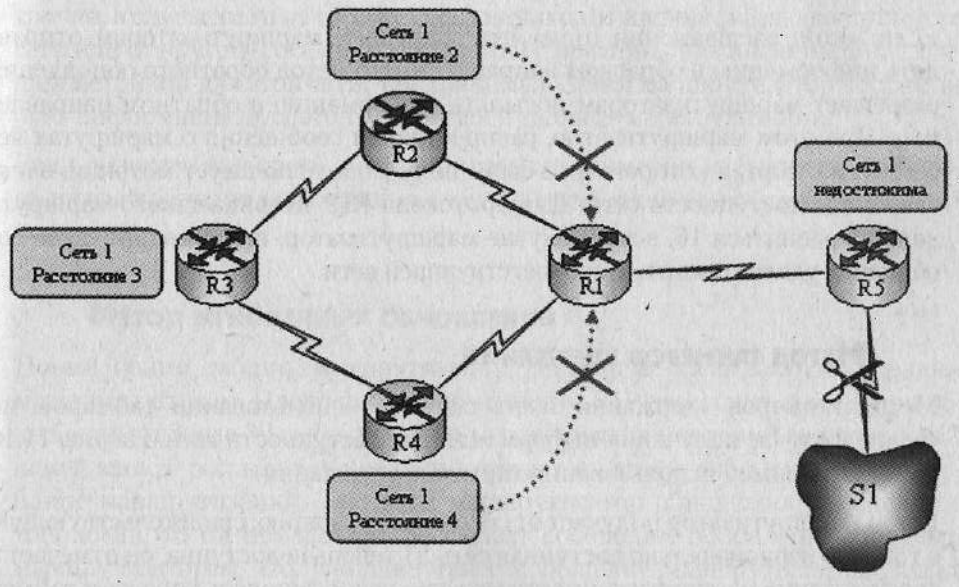


Рис. 11.9. Расщепление горизонта

Маршрутизатор R1 передаёт маршрутизаторам R2 и R4 пакет с обновлением маршрутной информации, говорящий о том, что сеть S1 стала недоступна. Однако маршрутизатор R3 передаёт маршрутизатору R4 пакет обновления, который информирует, что сеть S1 доступна по маршруту с расстоянием 4 через маршрутизатор R2. Маршрутизатор R4 посылает маршрутизатору R1 пакет обновления информации, в котором даёт совет R1 о "новом" маршруте к сети S1. Теперь маршрутизатор R1 определяет, что он может посылать пакеты в сеть S1 через маршрутизатор R4; маршрутизатор R4 определяет, что он может посылать пакеты через маршрутизатор R3, а маршрутизатор R3 определяет, что он может посылать пакеты в сеть S1 через маршрутизатор R2. Любой пакет, помещенный в такую среду, будет ходить по кругу между маршрутизаторами.

Метод расщепления горизонта заключается в том, что нет ничего хорошего в посылке информации о маршруте назад в направлении, из которого она первоначально была получена. Поэтому при поступлении сообщения об обновлении маршрутов для сети S1 от маршрутизатора R1 маршрутизаторы R2 и R4 не могут посылать информацию о сети S1 в обратном направлении. В результате чего информация о недоступности сети S1 распространяется только в одном направлении, и её постепенно получат все маршрутизаторы.

Метод обратного обновления

Если метод расщепления горизонта запрещает маршрутизаторам отправлять информацию в обратном направлении, то метод обратного обновления разрешает маршрутизаторам посылать информацию в обратном направлении. При этом маршрутизатор, распространяя сообщения о маршрутах через тот же порт, с которого они были получены, использует метрики, означающие недоступность сети. Для протокола RIP метрика такого маршрута должна равняться 16, в этом случае маршрутизатор, получивший такое сообщение, удаляет запись о соответствующей сети.

Метод таймеров удержания

Метод таймеров удержания заключается в использовании таймеров, но только в случае получения информации о недоступности сети. На рис. 11.10 приведён пример использования таймеров удержания.

Когда маршрутизатор получает от соседа информацию, свидетельствующую о том, что первоначально доступная сеть S1 теперь недоступна, он отмечает в своей таблице маршрутизации эту сеть как недоступную и запускает таймер удержания этой информации.

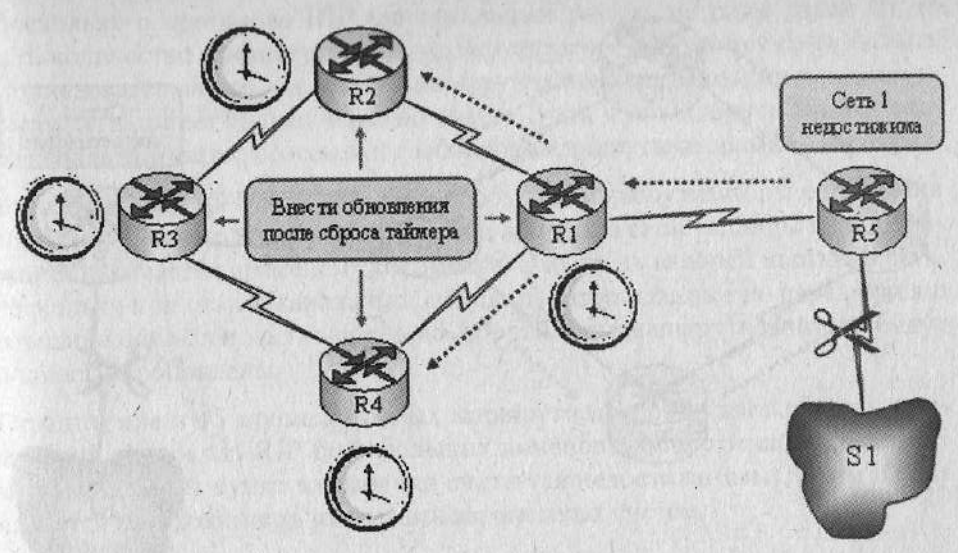


Рис. 11.10. Метод таймеров удержания

Сброс таймера осуществляется по достижении указанного времени или досрочно, в случае если от того же соседа приходит информация о доступности сети или от другого соседнего маршрутизатора приходит информация с лучшей метрикой для этой сети, чем хранимая. Затем маршрутизатор корректирует информацию для этой сети в своей таблице маршрутизации.

Если до истечения срока, устанавливаемого таймером удержания, от любого соседа приходит информация с худшей метрикой, то эта информация игнорируется.

Метод мгновенных обновлений

Новые копии таблиц маршрутизации регулярно рассылаются соседним маршрутизаторам. Протокол маршрутизации RIP рассылает сообщения обновлений каждые 30 секунд. Метод мгновенных обновлений заключается в немедленной рассылке обновлений в ответ на какое-либо изменение в таблице маршрутизации. Так, если маршрутизатор обнаружил изменение в топологии, то он немедленно рассылает сообщение-обновление смежным маршрутизаторам. Получившие такое сообщение маршрутизаторы также генерируют мгновенные обновления, оповещая о переменах своих соседей. Эта волна обновлений быстро распространяется по всей сети (рис. 11.11).

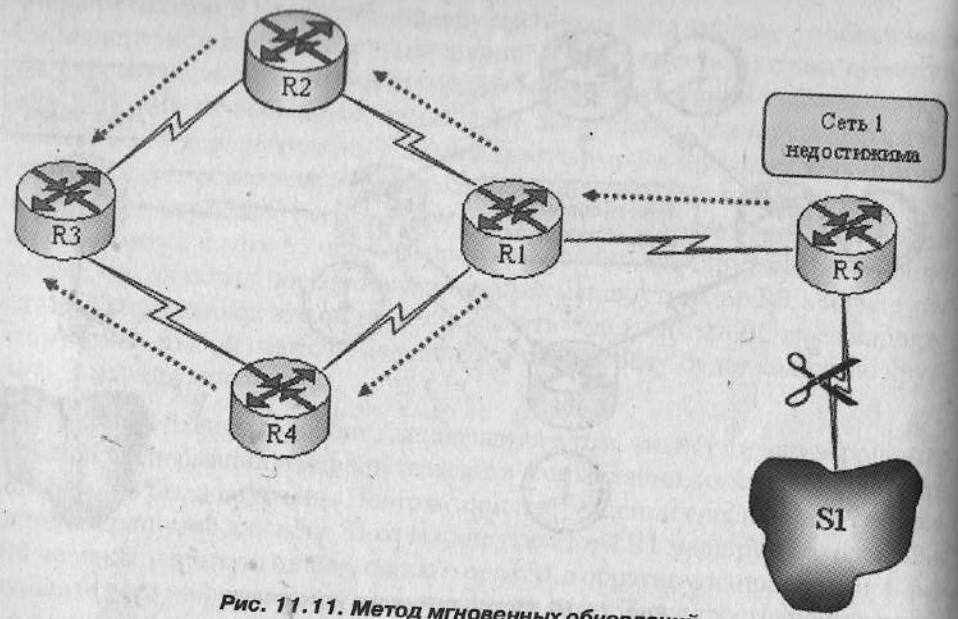


Рис. 11.11. Метод мгновенных обновлений

На рис. 11.11 маршрутизатор R5 генерирует мгновенное обновление, извещая о том, что сеть S1 недостижима. После получения этой информации маршрутизатор R1 извещает соседние с ним маршрутизаторы R2 и R4, а они в свою очередь маршрутизатор R3.

Приведённые примеры хорошо иллюстрируют главную причину нестабильной работы маршрутизаторов, работающих на основе дистанционно-векторного алгоритма, – использование информации, полученной из вторых рук. Искоренить эту причину полностью нельзя, так как сам способ построения таблиц маршрутизации связан с передачей чужой информации без указания на источник её происхождения.

В протоколе RIP в качестве времени удержания таймера выбрано шестикратное значение времени периода рассылки, то есть 180 секунд. Шестикратный запас времени необходим для того, чтобы удостовериться в том, что сеть действительно стала недоступной, а не произошли потери сообщений. Поэтому при выходе маршрутизатора из строя или канала связи все соседние с ним маршрутизаторы только через 180 секунд в своих таблицах маршрутизации все записи, порождённые этим маршрутизатором, сделают недействительными. После этого аналогичный процесс повторится уже для соседей ближайших соседей – они уничтожат подобные записи уже через 360 секунд, поскольку первые 180 секунд ближайшие соседи ещё передавали информацию об этих записях.

Поскольку в протоколе RIP максимальный размер метрики равен 15, то есть количество промежуточных маршрутизаторов не может быть больше 15, следовательно, до последнего маршрутизатора информация о недоступности сети дойдёт только через 45 минут. Этим и объясняется относительно небольшой период рассылки сообщений маршрутизаторами – 30 секунд.

Время, которое требуется для того, чтобы все маршрутизаторы обработали обновление маршрутной информации и обновили свои таблицы маршрутизации, называется временем сходимости. Это очень важный параметр сети, поскольку при отказе канала или маршрутизатора данные не передаются в объединенной сети до тех пор, пока все таблицы маршрутизации не будут полностью обновлены.

Ограничение в 15 промежуточных маршрутизаторов сужает область применения протокола RIP до небольших доменов маршрутизации (после 15 маршрутизатора пункт назначения считается недостижимым). Этим и объясняется необходимость в создании автономных систем.

11.2. Алгоритм кратчайшего пути

Алгоритм кратчайшего пути – алгоритм Дейкстры – работает с графами, состоящими из вершин, соединенных ребрами, количество которых может быть любым. Каждое ребро имеет стоимость и соединяет только две вершины в одном направлении. Например, если ребро соединяет вершину A с вершиной B, то это говорит о возможности перемещения из вершины A в вершину B за стоимость этого ребра. Для перемещения из вершины B в вершину A требуется наличие другого ребра.

На рис. 11.12 приведён пример графа. Этот граф содержит шесть вершин, отмеченных буквами от A до F. Ребра обозначаются линиями со стрелками, указывающими направление возможного перемещения, а стоимость ребер указана в кружочке на ребре.

Граф с большим количеством вершин может иметь множество путей между двумя вершинами. Среди этих путей лучшим считается путь, суммарная стоимость которого минимальна. Например, кратчайший путь от вершины D к вершине A пролегает через вершину B, при стоимости равной 4. Обратный путь, от вершины A к вершине D, имеет большую стоимость равную 15 и проходит через вершины C и B.

В приведённом примере задача нахождения кратчайшего пути относительно проста, но если граф становится больше, количество вычислений растет по экспоненте. И даже самый быстрый компьютер будет затрачивать слишком

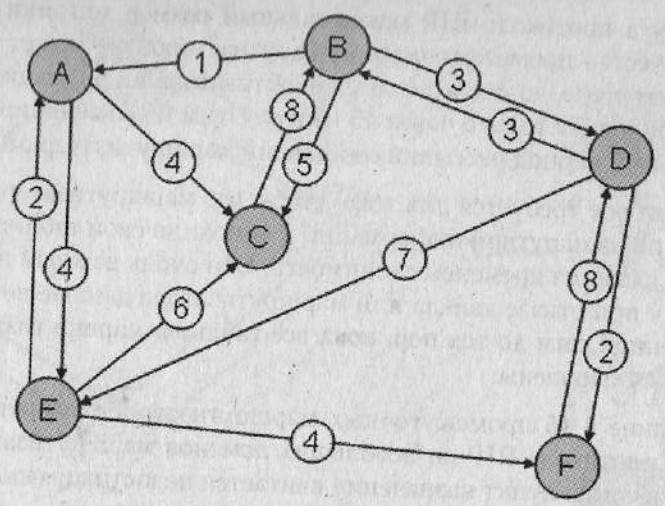


Рис. 11.12. Пример графа

много времени на перебор всех возможных путей и расчёт их стоимости. Алгоритм кратчайшего пути решает задачу быстрого нахождения самого короткого пути между любыми двумя вершинами в графе с произвольными связями.

В алгоритме используются две вспомогательные структуры данных: база данных вершин, для которых ищется кратчайший путь, и база данных вариантов вершин, для которых кратчайший путь может быть найден. Обозначаются эти базы как "найденные" и "кандидаты". Обе базы данных имеют идентичную структуру: они содержат два поля, содержащие ссылку на вершину и суммарную стоимость пути от начальной вершины к указанной вершине.

Чтобы продемонстрировать работу алгоритма по шагам, воспользуемся графом, изображенным на рис. 11.12. Работа алгоритма начинается с помещения начальной вершины в базу "найденные" с совокупной стоимостью 0, а всех её смежных вершин - в базу "кандидаты" с указанием суммарной стоимости до каждой из вершин, равной стоимости соответствующих ребер. После этого начинается циклическое выполнение следующих шагов:

- **Шаг 1**. Найти в базе "кандидаты" вершину с наименьшей суммарной стоимостью и переместить её в базу "найденные".
- **Шаг 2.** Определить вершины, с которыми перемещенная вершина является смежной.
- **Шаг 3.** Отбросить смежные вершины данной вершины, которые уже

были перемещены в базу "найденные".

- **Шаг 4.** Для каждой из смежных вершин перемещённой вершины, которая уже была помещена в базу "кандидаты", установить минимальную суммарную стоимость.
- **Шаг 5.** Поместить каждую смежную вершину, не содержащуюся ни в базе "найденные", ни в базе "кандидаты", в базу "кандидаты". Установить её суммарную стоимость как сумму стоимости перемещенной вершины и стоимости ребра.
- **Шаг 6.** Если база "кандидаты" пуста, завершить работу. В противном случае вернуться к Шагу 1 и повторить действия.

Попробуем найти кратчайшие пути до всех вершин с помощью алгоритма кратчайшего пути, например, от вершины А. На рис. 11.13 приведена последовательность шагов, которые необходимо выполнить.

Перед началом выполнения алгоритма определяется начальная вершина, в нашем примере это вершина А, данные о которой помещаются в базу «найденные». Естественно, что расстояние от вершины А к самой себе равно нулю.

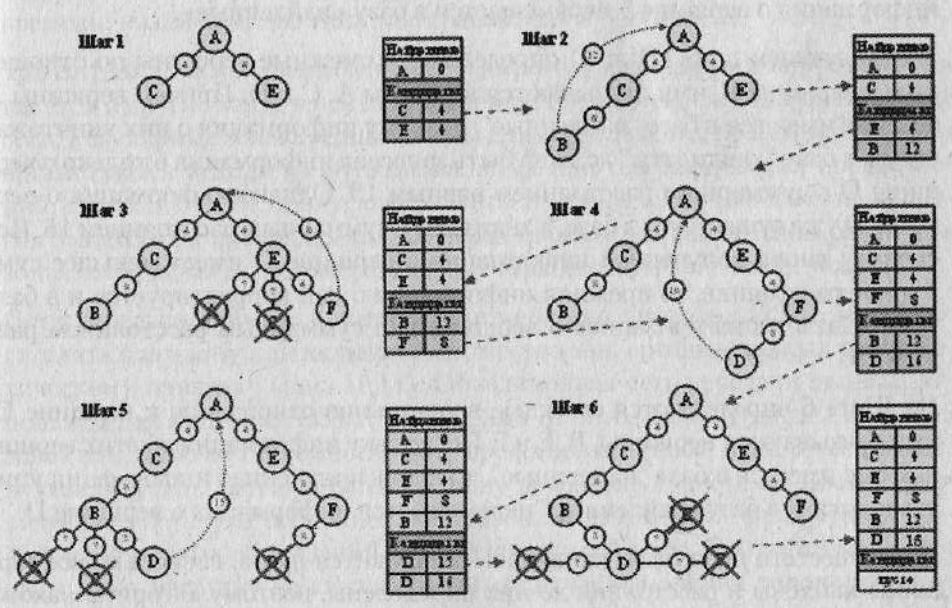


Рис. 11.13. Последовательность действий при нахождении кратчайшего пути

На первой шаге в базу «кандидаты» помещаются смежные вершины по отношению к вершине А с указанием стоимости пути до каждой из них соответственно – это вершины С и Е. Вершина В не является смежной, поскольку из вершины А нельзя попасть в вершину (направление ребра указывает перемещение в обратную сторону).

На следующем шаге (Шаг 2) необходимо определить смежные вершины по отношению к вершинам С и Е. В нашем примере определяются смежные вершины по отношению к вершине С. С вершиной С оказывается смежной только одна вершина В (см. рис. 11.12), имеющая суммарное расстояние равное 12. В связи с тем, что больше смежных вершин нет, то информация о вершине С перемещается в базу «найденные».

На следующем шаге (Шаг 3) определяются смежные вершины по отношению к вершине Е, ими оказываются вершины А, С и F. Причём вершины А и С уже имеются в базе «найденные», поэтому информация о них уничтожается, а в базу «кандидаты» заносится информация только о вершине F с суммарным расстоянием 8.

На Шаге 4 определяются смежные вершины по отношению к вершине F. Вершина F имеет только одну смежную вершину D с суммарным расстоянием равным 16, информация о которой и помещается в базу «кандидаты». А информация о вершине F перемещается в базу «найденные».

На следующем шаге (Шаг 5) определяются смежные вершины по отношению к вершине В, ими оказываются вершины А, С и D. Причём вершины А и С уже имеются в базе "найденные", поэтому информация о них уничтожается, а в базу "кандидаты" должна быть занесена информация о только о вершине D с суммарным расстоянием равным 15. Однако информация о вершине D уже существует в базе "кандидаты с суммарным расстоянием 16. Поскольку вновь полученная информация о вершине D имеет меньшее суммарное расстояние, то прежняя информация о ней корректируется и в базе "кандидаты" остаётся запись о вершине D с суммарным расстоянием равным 15.

На Шаге 6 определяются смежные вершины по отношению к вершине D, ими оказываются вершины В, Е и F. Поскольку информация об этих вершинах уже имеется в базе "найденные", то вновь полученная информация уничтожается, а в базу "найденные" перемещается информация о вершине D.

После шестого шага база "кандидаты" становится пуста, поскольку все вершины найдены и расстояния до них определены, поэтому алгоритм нахождения кратчайшего пути завершает свою работу.

Как видно из примера, алгоритм нахождения кратчайшего пути позволяет

определить кратчайшие пути к каждой вершине методично и достаточно просто.

Простота и скорость алгоритма определили для него центральное место в алгоритмах маршрутизации по состоянию канала, строящих маршруты исходя из информации о топологии сети.

11.3. Протокол OSPF

Алгоритм нахождения кратчайшего пути реализован в протоколе OSPF. В небольших сетях количество каналов связи межу маршрутизаторами не столь велико и расчёт маршрутов для каждой сети не очень сложен. Однако в больших сетях, где присутствует большое количество каналов связи между маршрутизаторами и количество возможных маршрутов может быть велико, выполнение алгоритма SPF требует достаточного много времени и значительной вычислительной производительности маршрутизатора. Для уменьшения количества расчётов в протоколе OSPF применяют разделение сети на зоны. Количество маршрутизаторов в каждой зоне не велико, следовательно, база данных о состоянии каналов в пределах зоны значительно меньше. Поэтому расчёт маршрутов становится проще и занимает меньше времени. Различают два типа зон: транзитная и регулярная.

Транзитная зона предназначена для быстрого и эффективного продвижения пакетов в другие зоны, в ней не рекомендуется размещать пользовательские сети. Регулярные зоны предназначены для подключения пользователей, они организуются исходя из функциональных или географических предпочтений, поэтому они не пропускают пакеты из других зон. Все пакеты из других зон должны проходить только через транзитную зону. Чтобы различать зоны, им присваивают номера. Главная (транзитная) зона имеет номер 0.

Сети передачи данных, в которых применяется протокол OSPF, могут составлять одну зону или включать множество зон, организованных по иерархическому признаку (рис. 11.14). Объединенная сеть передачи данных, использующая протокол OSPF, независимо от того, состоит ли она из одной зоны или включает множество зон, представляет собой один домен маршрутизации, или, другими словами, одну автономную систему. Такая иерархическая структура позволяет локализовать изменения маршрутов и трафик маршрутных обновлений в пределах каждой зоны. Соответственно, это уменьшает нагрузку на каналы связи, в случае изменения топологии сети. Для передачи информации между зонами используются пограничные маршрутизаторы, имеющие кратчайшее расстояние к зонам.

Все маршрутизаторы, работающие под управлением протокола OSPF, создают

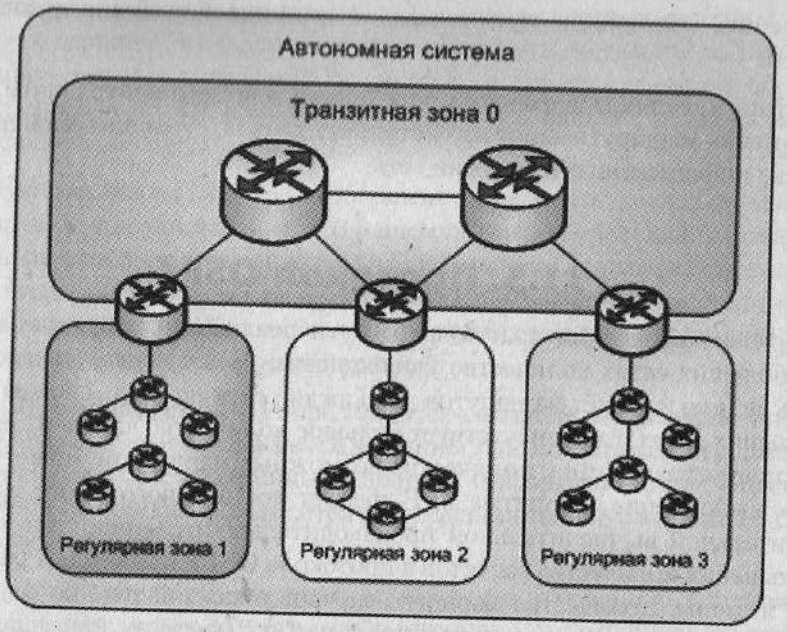


Рис. 11.14. Разделение сетей на зоны в протоколе OSPF

три таблицы: таблицу соседства и таблицу топологии, на основании которых строят таблицу маршрутизации.

В таблице соседства хранится список всех соседей и информация о состоянии связей с ними. Таблица топологии содержит информацию о состоянии всех сетей, подсетей и маршрутизаторов в пределах зоны, она строится на основании алгоритма нахождения кратчайшего пути. Если маршрутизатор подключён к двум и более зонам, то он создаёт отдельную таблицу топологии для каждой зоны. После завершения построения таблиц маршрутизации необходимо отслеживать изменения в топологии сети и вносить коррективы в таблицы маршрутизации.

Для поддержки таблицы соседства (отслеживания состояния всех своих соседей) в протоколе OSPF используется короткое сообщение Hello. Если состояние соседей не изменилось, то и перестраивать таблицу соседства не нужно. В случае изменения состояния связи маршрутизатор своим соседям посылает сообщение, касающееся только этой связи. Получив информацию об изменении состояния связи, маршрутизатор перестраивает граф и находит кратчайшие пути, причём не все, а только те, на которых отразилось изменение состояния связи, корректируя таблицу маршрутизации. Одновременно он ретранслирует объявление всем ближайшим соседям. Небольшой

объём этих сообщений и простая процедура их обработки позволяют передавать сообщения Hello достаточно часто.

В протоколе OSPF период рассылки сообщений Hello составляет 10 секунд. Получение от соседа сообщения Hello означает, что он продолжает функционировать. В некоторых случаях возможна задержка в передаче сообщения Hello, поэтому, чтобы маршрутизатор не решил, что сосед вышел из строя, он ожидает с четырёхкратным запасом – 40 секунд. Если в течение 40 секунд от соседа не поступает сообщение Hello, маршрутизатор решает, что тот гарантированно вышел из строя.

Для создания таблицы топологии соседние маршрутизаторы обмениваются известной им топологической информацией с помощью пакетов DBD. Обмен топологической информацией начинается после завершения установки соседских отношений. В качестве элементов топологии выступают маршрутизаторы, сети, суммарные маршруты и другая топологическая информация. Каждому элементу топологии соответствует уникальный тип сообщения – LSA. Каждому типу сообщения присваиваются номера. Это необходимо для того, чтобы при получении очередного сообщения об элементе топологии маршрутизатор изменял таблицу топологии только в случае получения действительно новой информации.

После получения пакета с топологической информацией от соседа маршрутизатор просматривает информацию об известных соседу элементах топологии и сравнивает её со своей таблицей топологии. Если информация об элементе топологии в полученном сообщении отсутствует в таблице, то он генерирует сообщение-запрос LSR для получения детальной информации об этом элементе топологии.

В ответ на запрос (сообщение LSR) сосед генерирует ответ – LSU, в котором находится затребованная информация. Получение каждого пакета LSU подтверждается отправкой пакета подтверждения. Если маршрутизатор в течение 5 секунд не дожидается подтверждения о получении пакета, то он производит повторную передачу. Параллельно с этим маршрутизаторы обмениваются сообщениями Hello. Таким образом, постепенно маршрутизаторы синхронизируют свои таблицы топологии, после чего производят запуск алгоритма нахождения кратчайшего пути.

Если топология сети меняется, то маршрутизаторы передают информацию об изменении. Изменение топологии может касаться состояния существующих связей, а может касаться состояния маршрутизатора (отказ или появление нового маршрутизатора).

Отказ связей определяется отсутствием сообщений Hello. Об отказе же или появлении нового маршрутизатора ближайшие соседи узнают по появлению

в составе пакета с топологической информацией новых записей (LSA), которые описывают изменения, произошедшие в топологических элементах сети. Причём эти сообщения рассылаются сразу же после обнаружения изменений топологии всем соседним маршрутизаторам, в свою очередь каждый маршрутизатор, получивший такое сообщение, производит обновление своей базы данных топологии сети и производит его дальнейшую рассылку всем своим соседям. Такая рассылка называется лавинной, и она информирует все маршрутизаторы о произошедших изменениях в топологии сети, а так же о возможной необходимости внесения изменений в таблицу маршрутизации.

В протоколе OSPF каждая запись в таблице топологии имеет срок жизни равный 30 минутам, поэтому каждые 30 минут маршрутизатор, у которого истекает срок жизни, должен запрашивать её новую копию непосредственно у маршрутизатора, породившего данную запись. Для корректной работы протокола OSPF необходимо учитывать не только время нахождения записи в таблице топологии, но и время, затраченное на его передачу по каналам связи. По умолчанию данный параметр равен 1 секунде.

Поскольку информация о связи генерируется исключительно тем маршрутизатором, который выяснил состояние этой связи, а остальные маршрутизаторы только транслируют эту информацию, то появление недостоверной информации о связи исключается.

Периоды нестабильной работы сети под управлением протокола OSPF возможны только в случае, когда информация об изменении сети ещё не достигла всех маршрутизаторов. Если пакеты невозможно доставить получателю, они просто уничтожаются, что исключает появление маршрутных петель. К недостаткам протокола OSPF следует отнести его вычислительную сложность и сложность настройки при реализации многозонной структуры сети.

11.4. Типы маршрутизаторов в протоколе OSPF

Протокол OSPF разграничивает функции маршрутизаторов в зависимости от того, какое место в домене маршрутизации они занимают и к какому числу зон принадлежат. На рис. 11.15 приводится пример домена маршрутизации OSPF, разделенного на зоны, и функций маршрутизаторов, которые они выполняют в зависимости от своего расположения.

Протокол OSPF использует четыре типа маршрутизаторов: внутренние, магистральные, пограничные зоны и пограничные автономной системы.

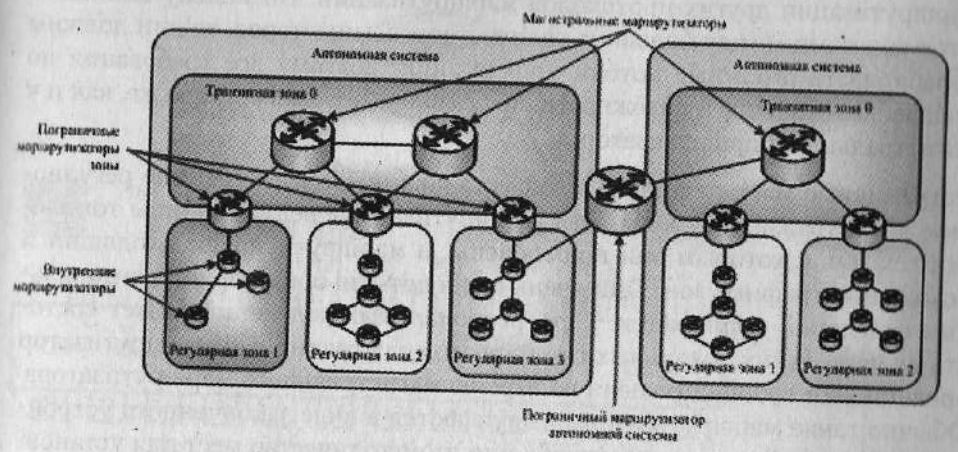


Рис. 11.15. Типы маршрутизаторов в протоколе OSPF

Поскольку автономная система может содержать большое количество регулярных зон, то такая конфигурация сети прекрасно подходит для организации сети корпорации, состоящей из большого количества локальных сетей, разбросанных по разным зданиям и использующих самые разнообразные сетевые технологии, типы компьютеров и операционных систем.

Магистральный маршрутизатор, находящийся в транзитной зоне, является центральной частью автономной системы – корпорации. Магистральные маршрутизаторы – это самые мощные устройства, способные обрабатывать сотни или даже тысячи миллионов пакетов в секунду, построенные на одном или нескольких специализированных RISC-процессорах, работающие под управлением специализированной операционной системы реального времени, имеющие большое количество портов, рассчитанных на использование самых высокоскоростных каналов связи (10 Гбит/с и выше). Чаще всего магистральные маршрутизаторы выполняются по модульной схеме. В таких маршрутизаторах большое внимание уделяется надёжности и отказоустойчивости, что достигается за счёт резервирования особо важных устройств с функцией горячей замены.

Пограничные маршрутизаторы автономных систем располагаются на границе двух и более автономных систем, в которых могут быть запущены различные протоколы маршрутизации. Основной функцией пограничных маршрутизаторов автономных систем является экспорт и импорт маршрутной информации между доменами маршрутизации OSPF и доменами

маршрутизации других протоколов маршрутизации. Поскольку они являются шлюзами между большим количеством компьютеров, то они должны обрабатывать огромные потоки информации, поэтому все требования по производительности и надёжности, предъявляемые к ним, такие же, как и к магистральным маршрутизаторам.

Пограничные маршрутизаторы зон устанавливаются на границе регулярных зон и транзитной зоны. Эти маршрутизаторы ведут таблицы топологии тех зон, к которым они подключены, и маршрутизируют входящий и исходящий трафики зон. Они очень подходят для организации региональных отделений корпорации. Сеть региональных отделений может состоять из нескольких локальных сетей, поэтому пограничный маршрутизатор представляет собой упрощённую версию магистрального маршрутизатора. Обычно такие маршрутизаторы выпускаются в виде законченного устройства или в модульном виде, однако при этом количество мест для установки модулей невелико (4-8). Это наиболее обширный класс выпускаемых маршрутизаторов, характеристики которых лежат в широком диапазоне – они могут приближаться к характеристикам магистральных маршрутизаторов, а могут опускаться до характеристик внутренних маршрутизаторов зоны.

Внутренние маршрутизаторы – это маршрутизаторы, все порты которых находятся в одной зоне протокола OSPF, они имеют идентичные таблицы топологии данной зоны. С помощью этих маршрутизаторов сети конечных пользователей связываются между собой. Они идеально подходят в качестве маршрутизаторов удалённых офисов. Как правило, они соединяют одну или несколько локальных сетей удалённого офиса с центральной сетью или сетью регионального отделения. Существует очень большое количество типов маршрутизаторов удалённых офисов. Это объясняется как массовостью потенциальных потребителей, так и специализацией, выражающейся в поддержке конкретного типа связи.

11.5. Коммутаторы 3-го уровня

Коммутаторы третьего уровня предназначены для разделения локальных сетей на подсети с помощью организации виртуальных сетей. Поскольку передача информации между виртуальными сетями на уровне физического адреса запрещена, то для организации связи между виртуальными сетями необходимо использовать 3-й уровень модели OSI и соответственно специальное устройство, работающее на этом уровне, – маршрутизатор.

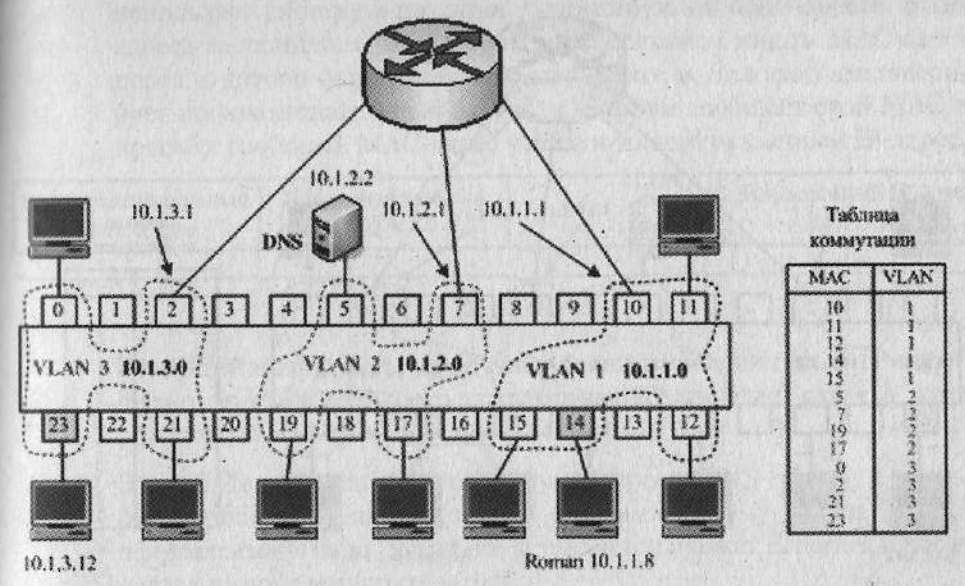


Рис. 11.16. Организация виртуальных сетей на основе портов коммутатора

Коммутация между виртуальными сетями на внешнем маршрутизаторе

Если коммутатор поддерживает несколько виртуальных сетей, но не имеет возможности работать на уровне 3 модели OSI, то он должен быть подключён к маршрутизатору с помощью линий связи, причём каждая виртуальная сеть должна иметь индивидуальную линию связи. Поскольку виртуальные сети на коммутаторах могут быть реализованы на основе портов или MAC-адресов, то в таблице коммутации коммутатора необходимо указать соответствующие адреса портов коммутатора (рис. 11.16) или MAC-адреса портов всех подключаемых узлов (рис. 11.17). Для передачи пакетов между виртуальными сетями, построенными на нескольких коммутаторах, необходимо использовать специализированный протокол, позволяющий коммутатору указывать принадлежность пакета соответствующей виртуальной сети.

Независимо от того, на каком принципе организованы виртуальные сети, процесс передачи информации между ними состоит из нескольких этапов. Для пояснения процесса маршрутизации пакетов между виртуальными сетями рассмотрим пример, изображённый на рис. 11.16. Коммутатор работает в стандарте Ethernet. В данном случае узел, подключенный к порту 23 коммутатора и принадлежащий виртуальной сети VLAN 3, желает передать информацию узлу, подключённому к порту 14 коммутатора, находящемуся в

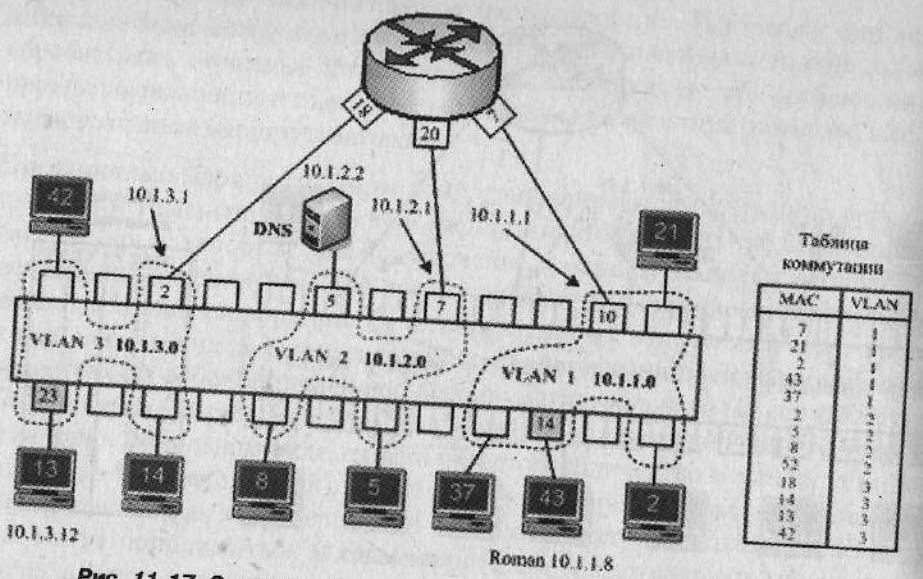


Рис. 11.17. Организация виртуальных сетей на основе MAC-адресов

виртуальной сети VLAN 1. Поскольку маршрутизатор работает со своей собственной системой адресации, то администратор должен организовать три сети, каждой из них присвоить IP-адрес и в каждой сети распределить адреса между узлами. В нашем примере используются сети с номерами 10.1.1.0; 10.1.2.0 и 10.1.3.0. Порты, к которым подключен маршрутизатор, во всех сетях имеют одинаковый номер – 0.0.0.1. Узлам, подключенным к портам 23 и 14, достались адреса 10.1.3.12 и 10.1.1.8.

Как отмечалось ранее, на каждом узле должно быть установлено программное обеспечение стека TCP/IP. При настройке системы администратор заносит в таблицу маршрутизации каждого узла несколько записей, которые в частности включают: записи о непосредственно подключённых к узлу сетях и маршрутизаторах, а также запись о ближайшем DNS-сервере (в нашем примере он находится в VLAN 2 с адресом 10.1.2.2).

- **Этап 1**. Узел, находящийся в сети VLAN 3, подключённый к порту 23 коммутатора (IP-адрес которого 10.1.3.12), желает передать информацию узлу по имени Roman, находящемуся в другой виртуальной сети. Чтобы передать информацию в другую сеть, необходимо указать IP-адрес узла по имени Roman, однако он его не знает. Поэтому узел должен обратиться к DNS-серверу за помощью. Обратиться к DNS-серверу можно, но только через маршрутизатор, IP-адрес которого имеется в таблице маршрутизации узла. Однако коммутатор

использует систему адресации, основанную на MAC-адресе, и IP-адреса не понимает. Поэтому наш узел должен узнать MAC-адрес порта, к которому подключен маршрутизатор. Для этого он генерирует широковещательный запрос, в котором сообщает свой MAC и просьбу: сообщить MAC-адрес узла, имеющего указанный IP-адрес.

Широковещательный MAC адрес	Адрес отправителя 23		Запрошенный IP адрес 10.1.3.1.

Кадр Ethernet

- **Этап 2**. Маршрутизатор получает пакет, распознаёт свой IP-адрес, читает просьбу и в адрес узла отправителя посылает ответ, в котором указывает MAC-адрес своего порта – 2.
- **Этап. 3**. Теперь наш узел генерирует запрос к DNS-серверу с просьбой сообщить IP-адрес узла по его символьному имени. Запрос оформляется в виде IP-пакета, вставляется в пакет Ethernet и посылается на порт маршрутизатора.

Адрес получателя 2		IP адрес назначения 10.1.2.2		Запрошенный IP адрес Roman

Заголовок кадра Ethernet | Заголовок IP пакета | Поле данных IP пакета

- **Этап 4**. Маршрутизатор получает кадр Ethernet, извлекает из него IP-пакет и определяет, что данный пакет необходимо переслать в сеть 10.1.2.0 узлу 0.0.0.5 через порт 7 коммутатора. Если маршрутизатор ещё ни разу не обращался к DNS-серверу, то он не знает его MAC-адреса. Тогда он должен послать широковещательный запрос в сеть с номером 10.1.2.0 и попросить узел с адресом 10.1.2.5 сообщить свой MAC-адрес. После чего он может транслировать полученный пакет DNS-серверу.
- **Этап 5**. После получения пакета и извлечения из него запроса DNS-сервер начинает просмотр своих баз данных. Если информация об указанном пользователе Roman найдена, то он её отправляет маршрутизатору. Если искомая информация отсутствует, то он должен обратиться к вышестоящему DNS-серверу, IP-адрес которого у него имеется. Будем считать, что искомая информация – IP-адрес 10.1.1.8 – найдена. Маршрутизатор пересылает полученную информацию узлу назначения по MAC-адресу 2.

- **Этап 6**. Наконец узел 10.1.3.12 получает IP-адрес узла назначения по имени Roman. Теперь он генерирует IP-пакет, в который помещает IP-адрес назначения и данные для него, вставляет его в кадр Ethernet и отправляет на порт 2 коммутатора маршрутизатору.
- **Этап 7**. Маршрутизатор из полученного кадра извлекает IP-пакет и определяет, в какую сеть, через какой порт коммутатора необходимо послать информацию. При этом он должен знать MAC-адрес получателя. Если он его не знает, то должен сгенерировать широковещательный запрос узлу с IP-адресом 10.1.1.8 с просьбой откликнуться и сообщить свой MAC-адрес. Только теперь маршрутизатор может ретранслировать полученную информацию узлу с именем Roman.

Если узел с адресом 10.1.3.12 пожелает послать ещё несколько пакетов узлу по имени Roman, то маршрутизатор, зная адреса всех участников (IP и MAC), всё равно для каждого пакета будет выполнять процедуру маршрутизации, которая включает: распаковывание кадров Ethernet, извлечение IP-адреса назначения, осуществление просмотра своей таблицы маршрутизации и определение MAC-адреса узла назначения.

Использование внешнего маршрутизатора имеет свои преимущества: простота исполнения, не требующая реализации функций маршрутизации на коммутаторе. Однако применение внешнего маршрутизатора приводит к увеличению времени прохождения пакета за счёт затрат времени на определение маршрута, а отказ последнего приводит к невозможности обмена информацией между виртуальными сетями. Кроме этого, при интенсивном обмене информацией между виртуальными сетями и недостаточной пропускной способности каналов связи с маршрутизатором могут возникнуть дополнительные задержки.

Коммутатор с маршрутизацией виртуальных сетей

Традиционно коммутатор принимает решения о передаче пакета на основе физической информации (физический адрес устройства), содержащейся в заголовке пакета, тогда как маршрутизатор руководствуется логическими адресами.

Коммутатор с возможностью маршрутизации пакетов, реализующий механизмы маршрутизации 3-го уровня модели OSI, принято называть коммутатором 3 уровня или многоуровневым коммутатором, а метод, используемый для коммутации, принято называть многоуровневой коммутацией – MLS соответственно. В технической литературе разных авторов и компаний используются оба определения.

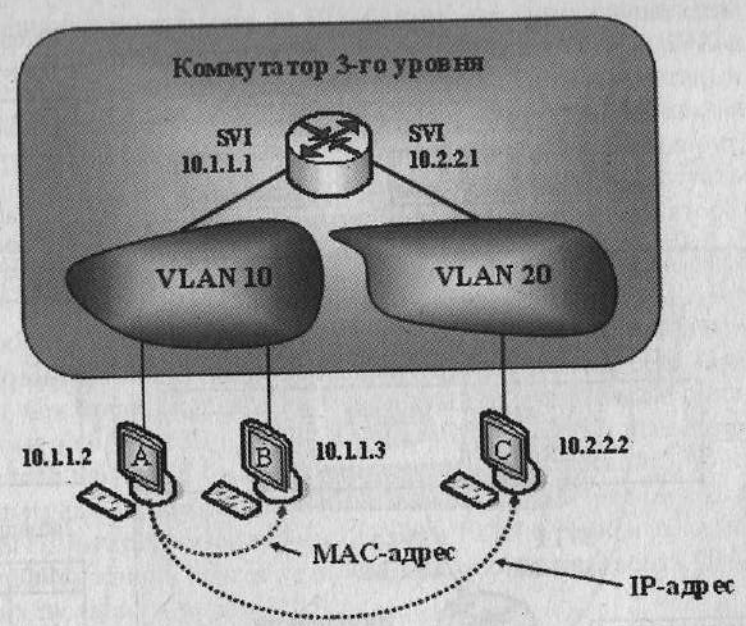


Рис. 11.18. Организация передачи между виртуальными сетями

На рис. 11.18 приведён пример использования коммутатора 3-го уровня. Узлы A и B объединены в одну виртуальную сеть, поэтому передача пакетов между ними будет осуществляться на основе физических (MAC) адресов, используемых на 2-м уровне модели OSI, а передача пакетов между виртуальными сетями осуществляется с использованием логических адресов, применяемых на 3-м уровне указанной модели. Многоуровневые коммутаторы передают пакеты со скоростью передающей среды, используя специальные встроенные в коммутатор аппаратные средства, которые реализуют функции маршрутизации, выполняемые ранее только маршрутизаторами. Для выбора направления передачи кадра на обоих уровнях используются собственные таблицы соответствия, только на уровне 3 они называются таблицами маршрутизации и хранятся в памяти TCAM, а на уровне 2 – таблицами коммутации. Кроме указанных таблиц для осуществления быстрой маршрутизации используется временная таблица, которая называется CAM.

Для пояснения процесса маршрутизации на коммутаторе 3-го уровня обратимся к рис. 11.19. Коммутатор 3-го уровня состоит из двух устройств: собственно коммутатора – MLS-SE, способного создавать виртуальные сети, и маршрутизатора – MLS-RP. Перед запуском в работу такого устройства необходимо произвести настройку как коммутатора, так и маршрутизатора.

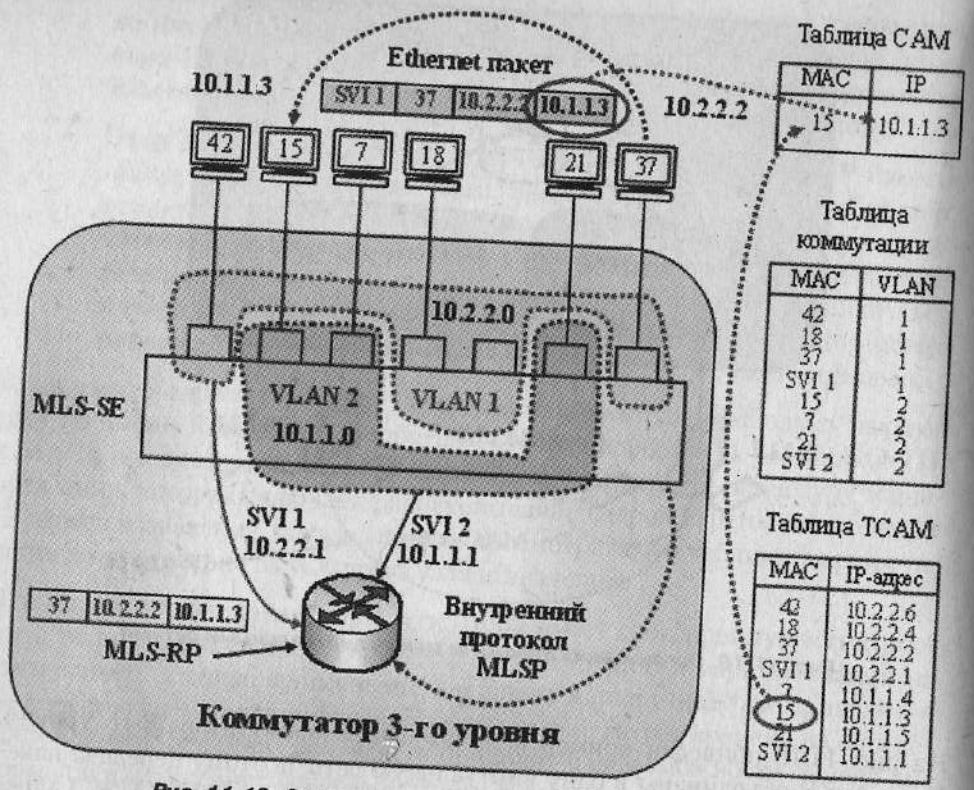


Рис. 11.19. Организация передачи внутри коммутатора

Первое, что нужно сделать, – это организовать виртуальные сети статические – на основе портов или динамические – на основе MAC-адресов (зависит от возможностей коммутатора). В приведённом примере используется коммутатор, реализующий динамические виртуальные сети. Поэтому администратор, создавая виртуальные сети, заполняет таблицу коммутации. Поскольку маршрутизатор является неотъемлемой частью коммутатора, то для организации связи с маршрутизатором необходимо создать виртуальный интерфейс коммутации – SVI. Индивидуальные виртуальные интерфейсы коммутации SVI должны быть созданы для каждой виртуальной сети, подлежащей маршрутизации, они выполняют функции физического порта маршрутизатора и настраиваются как физические.

По умолчанию на всех коммутаторах уже существует одна виртуальная сеть VLAN1 и один виртуальный интерфейс коммутации SVI1 для неё. Дополнительные виртуальные интерфейсы коммутации создаются администратором.

В приведённом примере (рис. 11.19) созданы две виртуальные сети VLAN1 и VLAN2 и два виртуальных интерфейса коммутации SVI1 и SVI2 для каждой виртуальной сети соответственно. Создана таблица коммутации, в которой указана принадлежность узлов виртуальным сетям. Так как виртуальные интерфейсы выполняют функции физических портов маршрутизатора, то информация об их принадлежности соответствующим виртуальным сетям тоже заносится в таблицу коммутации. Для связи коммутатора с маршрутизатором используется внутренний протокол многоуровневой коммутации – MLSP.

Поскольку маршрутизатор использует свою собственную систему адресации, основанную на IP-адресах, то каждой виртуальной сети необходимо выделить некоторое адресное пространство в соответствии с принципами, принятыми в системе IP-адресов. Поскольку эти адреса используются исключительно для нужд коммутатора (внутренняя адресация), то их можно выбирать из любого диапазона любого класса. В приведённом примере виртуальным сетям присвоены адреса 10.2.2.0 и 10.1.1.0. Чтобы узлы могли предавать информацию в другие сети, им необходимо присвоить IP-адреса из диапазона соответствующих сетей.

В результате создаётся таблица маршрутизации – TCAM, в которой каждому MAC-адресу узла ставится в соответствие его IP-адрес. Поскольку виртуальные интерфейсы коммутации выполняют функции портов маршрутизатора, то им тоже присваиваются IP-адреса из диапазона адресов, присвоенного соответствующей виртуальной сети.

Для нормальной работы тандема коммутатор-маршрутизатор с помощью протокола многоуровневой коммутации маршрутизатор каждые 15 секунд отправляет групповые сообщения приветствия всем коммутаторам в сети. Эти сообщения информируют коммутаторы, что маршрутизатор является доступным и готов предоставить информацию маршрутизации коммутаторам.

На каждом узле должно быть установлено программное обеспечение стека TCP/IP. При настройке системы администратор заносит в таблицу маршрутизации каждого узла несколько записей, которые в частности включают: записи о непосредственно подключённых к узлу сетях и маршрутизаторах. Запись о ближайшем DNS-сервере обычно не используется, поэтому администратор должен записать адресную информацию обо всех устройствах, подключённых к коммутатору, в таблицу маршрутизации. В противном случае необходимо создать ещё одну виртуальную сеть, в которой будет находиться DNS-сервер.

Допустим, узел с MAC-адресом 37, принадлежащий VLAN1, желает послать сообщение узлу с MAC-адресом 15, который входит в состав VLAN2.

Процесс передачи информации из одной сети в другую принципиально аналогичен рассмотренному ранее процессу с использованием внешнего маршрутизатора. Однако существуют некоторые отличия, на которых стоит остановиться более подробно. Будем считать, что IP-адрес узла назначения известен отправителю.

Для передачи информации узел с IP-адресом 10.2.2.2 (MAC-адрес 37) формирует IP-пакет, в котором указывает IP-адрес получателя – 10.1.1.3 (MAC-адрес 15). Помещает его в стандартный Ethernet-кадр и посылает на виртуальный порт маршрутизатора SVI1 через физический порт, к которому подключён. Порт коммутатора воспринимает этот кадр, извлекает из него IP-пакет и переписывает IP-адрес получателя в таблицу CAM с отметкой "предварительная", а сам пакет сохраняет в своём буфере, затем транслирует заголовок IP-пакета на виртуальный порт маршрутизатора. Маршрутизатор получает пакет, запоминает адрес отправителя, анализирует содержимое своей таблицы маршрутизации – TCAM и определяет, что IP-адресу 10.1.1.3 соответствует MAC-адрес 15. Эту информацию он и отправляет в обратном направлении.

Порт коммутатора получает от маршрутизатора информацию о том, что IP-адресу 10.1.1.3 соответствует узел с MAC-адресом 15. Порт коммутатора извлекает из буфера IP-пакет, исправляет контрольную сумму, переписывает заголовок, вставляет IP-пакет в кадр Ethernet и отправляет его на порт 15. При этом в таблицу CAM добавляет информацию о том, что IP-адресу 10.1.1.3 соответствует MAC-адрес 15, и делает пометку "истинная".

Теперь любые дальнейшие пакеты, которые принадлежат тому же самому получателю, обрабатываются коммутатором и пересылаются на основании данных таблицы CAM (осуществляется поиск MAC-адреса по IP-адресу). Коммутатор только переписывает заголовки, исправляет контрольные суммы и отправляет пакеты самостоятельно без участия маршрутизатора, так, как будто они были отправлены маршрутизатором. Когда диалог между этими двумя устройствами заканчивается или прекращается по любой другой причине, запись в таблице CAM устаревает и удаляется. Для того чтобы мог состояться новый диалог, процесс должен начаться снова.

В современных коммутаторах размер таблицы CAM позволяет хранить до 1024 записей, что соответствует максимальному числу узлов в сети Ethernet и может достигать 128 Кбайт.

После того как два узла закончили обмен пакетами, информацию из таблицы коммутации можно удалить, причём, с одной стороны, чем быстрее будет обнаружен конец обмена и удалена из памяти запись о нём, тем более эффективно будет использовано пространство таблицы. С другой стороны, при

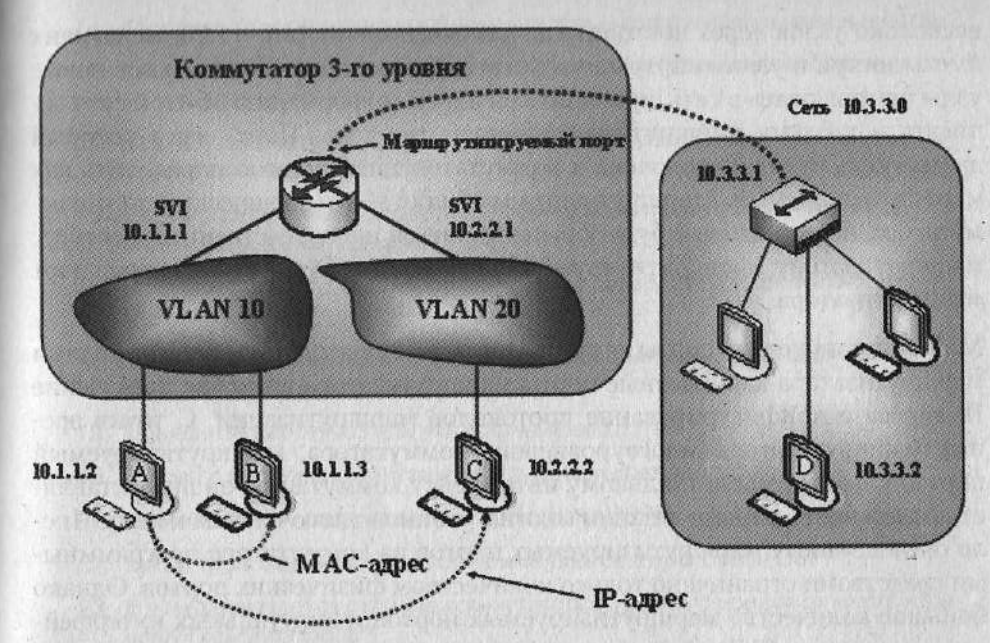


Рис. 11.20. Организация подключения внешней сети

возникновении необходимости передачи информации в направлении узлов, с которыми уже была связь, использование таблицы CAM позволит обойтись без услуг маршрутизатора, что скажется на повышении производительности.

Для удаления информации из таблицы CAM коммутаторы используют таймеры. Значение времени таймера устаревания может быть установлено в широких пределах от нескольких секунд до нескольких десятков минут, причём можно указать количество пакетов, которые должны быть обнаружены за данный интервал времени от 0 до 63. Если за указанный промежуток времени было передано пакетов больше заданного количества, то после завершения работы таймера запись из таблицы не удаляется, а таймер запускается вновь.

Правильно выбранные значения таймеров и количество пакетов при размере таблицы в 128 Кбайт обеспечивают хранение записей таблицы обычно до 256 секунд, если нет движения пакетов по данному направлению. Малое значение таймера времени устаревания целесообразно использовать для записей одноразового использования, типа поиска DNS.

Иногда бывает необходимо к коммутатору третьего уровня подключить ещё

несколько узлов через повторитель, как показано на рис. 11.20, желающих иметь доступ к узлам виртуальных сетей. Для этого необходимо все новые узлы организовать в сеть, присвоить ей адрес, информацию об этой сети записать в таблицу маршрутизации маршрутизатора. Порт, через который данная сеть будет подключена к коммутатору, необходимо определить как маршрутизируемый. Маршрутизируемый порт – это физический порт коммутатора, через который будет осуществляться непосредственный доступ к маршрутизатору. Маршрутизируемый порт ведёт себя как стандартный порт маршрутизатора.

Маршрутизируемые порты организуются и конфигурируются командами маршрутизатора как обычные порты маршрутизатора, включая назначение IP-адреса и конфигурирование протоколов маршрутизации. С точки зрения маршрутизатора многоуровневого коммутатора, маршрутизируемый порт эквивалентен виртуальному интерфейсу коммутации, он предоставляет возможность маршрутизации многим узлам, подключённым к нему. Число организуемых маршрутизируемых портов на коммутаторе программными средствами ограничено только количеством физических портов. Однако большое количество маршрутизируемых портов и виртуальных интерфейсов коммутации SVI, организованных на многоуровневом коммутаторе, может привести к перегрузке его аппаратных средств.

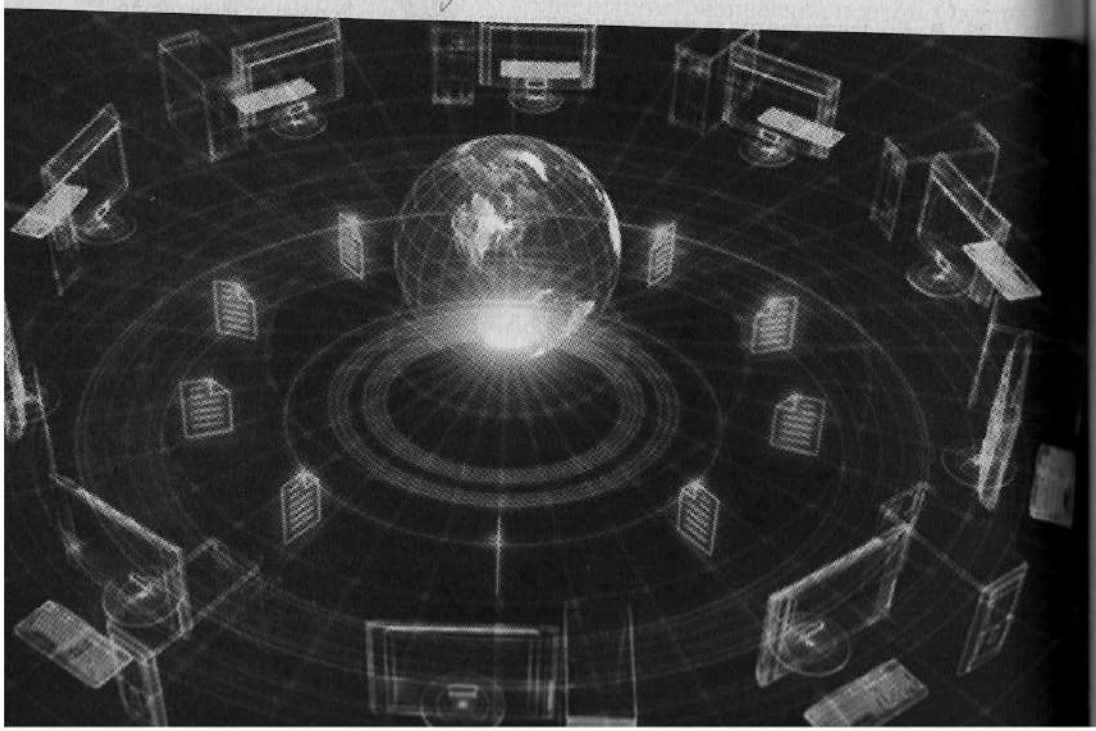
Контрольные вопросы

1. *В чём заключается основное назначение маршрутизатора?*
2. *Чем отличается маршрутизирующий протокол от маршрутизируемого протокола?*
3. *Объясните назначение таблицы маршрутизации.*
4. *В каких случаях целесообразно применение статической маршрутизации?*
5. *В чём отличие динамической маршрутизации от других способов маршрутизации?*
6. *Дайте определение автономной системе.*
7. *Дайте определение домена маршрутизации.*
8. *Назовите основное назначение протокола внешнего шлюза.*
9. *Какие алгоритмы маршрутизации вам известны?*
10. *Поясните назначение алгоритмов маршрутизации.*
11. *Объясните работу дистанционно-векторного алгоритма.*
12. *Поясните механизм возникновения петель маршрутизации.*

13. *Какие методы решения проблемы петель маршрутизации вам известны?*
14. *Каким образом метод счёта до бесконечности позволяет избавиться от петель маршрутизации?*
15. *За счёт чего метод расщепления горизонта позволяет избавиться от петель маршрутизации?*
16. *Каким образом метод обратного обновления позволяет избавиться от петель маршрутизации?*
17. *За счёт чего метод таймеров удержания позволяет избавиться от петель маршрутизации?*
18. *Каким образом метод мгновенных обновлений позволяет избавиться от петель маршрутизации?*
19. *Объясните работу алгоритма кратчайшего пути.*
20. *Для чего используются зоны в алгоритмах кратчайшего пути?*
21. *Какие типы зон реализованы в протоколе OSPF?*
22. *Какие типы маршрутизаторов применяются в протоколе OSPF?*
23. *В чём заключается отличие коммутаторов третьего уровня от многоуровневых коммутаторов?*
24. *Назовите основные компоненты многоуровневого коммутатора.*
25. *Опишите процесс работы многоуровневого коммутатора.*
26. *Каким образом можно осуществить передачу пакетов между виртуальными сетями?*
27. *Объясните достоинства использования внешнего маршрутизатора для коммутации виртуальных сетей.*
28. *Каким образом коммутатор 3-го уровня реализует процесс маршрутизации между виртуальными сетями?*
29. *Каким образом коммутатор 3-го уровня реализует процесс маршрутизации между виртуальными сетями и внешней сетью?*
30. *Поясните назначение виртуальных интерфейсов коммутации.*

Глава 12.

ПРОТОКОЛЫ TCP И UDP



Протоколы сетевого уровня обеспечивают доставку данных между узлами в составной сети (компьютерами, коммутаторами, маршрутизаторами и т.д.). Однако на одном узле могут параллельно функционировать несколько программ, которым требуется доступ к сети. Следовательно, данные внутри компьютерной системы должны распределяться не только между узлами, но и между программами. Поэтому при передаче данных по сети недостаточно просто адресовать конкретный узел, необходимо ещё идентифицировать программу-получатель, что невозможно осуществить средствами сетевого уровня.

Другой серьезной проблемой протоколов сетевого уровня является отсутствие средств, позволяющих передавать большие массивы данных. Когда исходные данные превышают максимально допустимый размер пакета сетевого уровня, то эти данные должны быть разбиты на порции, каждая из которых передается в сеть отдельным пакетом. Однако каждый пакет сетевого уровня передается по сети как единый, независимый от других блок данных. В случае если какие-либо пакеты «потерялись», то модуль сетевого протокола на принимающей стороне не сможет обнаружить потерю, и, следовательно, – обнаружить нарушение целостности общего массива данных. Поэтому необходимо каким-то образом обеспечить отсутствие потери информации. Такой режим передачи данных получил название гарантированной доставки.

Таким образом, протоколы транспортного уровня представляют собой функциональную надстройку над протоколами сетевого уровня и решают две основные задачи:

- обеспечение доставки данных между конкретными программами, функционирующими, в общем случае, на разных узлах сети;
- обеспечение гарантированной доставки массивов данных произвольного размера.

В настоящее время в Интернет используются два транспортных протокола – UDP, обеспечивающий негарантированную доставку данных между программами, и TCP, обеспечивающий гарантированную доставку с установлением виртуального соединения.

При описании различных протоколов передачи информации в сетях в технической литературе часто встречаются такие понятия, как **дейтаграмма, па-**

студентов вносит полную неразбериху. Поэтому прежде чем приступать к описанию протокола TCP, следует разобраться в этих понятиях.

Итак, понятие дейтаграмма происходит от английского слова datagram, которое на русский язык перевести крайне сложно, поскольку первая часть обозначает «данные, факты, информация», а вторая часть является компонентом сложных слов, например телеграмма, диаграмма, криптограмма, поэтому его используют без перевода в записи русскими буквами. Исходя из этого, можно говорить лишь о логическом значении этого слова, которое в иностранной технической литературе чаще всего обозначает сообщение, например, программа генерирует сообщение.

Слово **пакет** в русском языке обычно обозначает свёрток или письмо, адресованное кому-то. Поэтому применительно к сетям пакет обозначает сообщение, адресованное конкретному получателю.

Слово **фрейм** (frame) в переводе с английского языка означает костяк, остов, каркас. Применительно к нашему случаю, оно занимает промежуточное положение между сообщением и пакетом, поэтому его можно интерпретировать как группу пакетов или часть сообщения, в связи с этим наиболее близким синонимом является **блок данных**.

В дальнейшем мы будем использовать понятие блок данных для обозначения порции информации, **пакет** – в качестве сообщения, снабжённого адресами получателя и отправителя, а вместо слова **дейтаграмма** – более привычное нам – **сообщение**. Чтобы отличать пакеты протокола IP от пакетов протокола TCP, будем в дальнейшем называть пакеты протокола TCP – **сегментами**.

Протокол IP – это рабочая лошадка многих протоколов, которые используют его для передачи своих данных в сети, но он является ненадёжным протоколом, поскольку не обладает механизмами, обеспечивающими гарантированную доставку пакетов. Эту задачу и решает протокол более высокого уровня – TCP.

Протокол TCP (Transmission Control Protocol) работает над IP-протоколом, устанавливая логическое соединение между процессами, функционирующими на разных компьютерах, и использует для транспортировки своих пакетов ненадёжный протокол IP.

Если протокол IP обеспечивает передачу данных между любыми узлами сети, то протокол TCP осуществляет передачу информации между любыми прикладными задачами (логическими процессами), выполняющимися на любых узлах сети. Надёжность передачи данных протоколом TCP достигается наличием в его составе специальных средств, в результате ошибки,

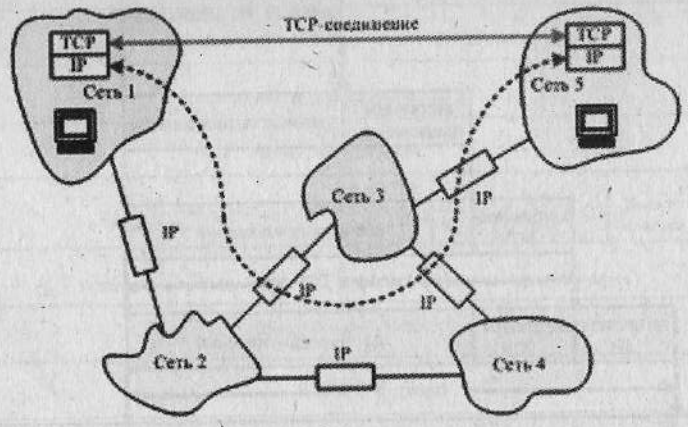


Рис. 12.1. TCP-протокол создаёт надёжный канал связи между узлами

возникающие на уровне IP-протокола, не влияют на правильное получение данных до тех пор, пока составная сеть не распадётся на несвязанные части (рис. 12.1).

После того, как пакет средствами IP-протокола доставлен узлу-получателю, полученные данные направляются конкретному процессу-получателю. Адресом назначения сегментов данных служат точки входа прикладного процесса, представляющие собой идентификатор (номер порта) прикладного процесса (на каждом компьютере может выполняться несколько процессов, а каждый процесс может иметь несколько точек входа).

Перед отправкой своих сообщений протокол TCP помещает их в пакеты IP-протокола, как показано на рис. 12.2, в соответствии с международной моделью взаимодействии открытых систем OSI (см. главу 1).

Протокол TCP имеет собственную систему адресации прикладных процессов. TCP-сегмент состоит из заголовка и поля данных. Весь сегмент представляется в виде набора 32-х разрядных слов, как показано на рис. 12.3. В первом слове заголовка сегмента располагаются адреса (номера портов) источника и приёмника.

Номера портов играют роль адресов транспортного уровня, идентифицируя на конкретных узлах сети потребителей транспортных услуг. Своя собственная адресация на транспортном уровне необходима для обеспечения возможности функционирования на узле сети одновременно многих сетевых приложений.

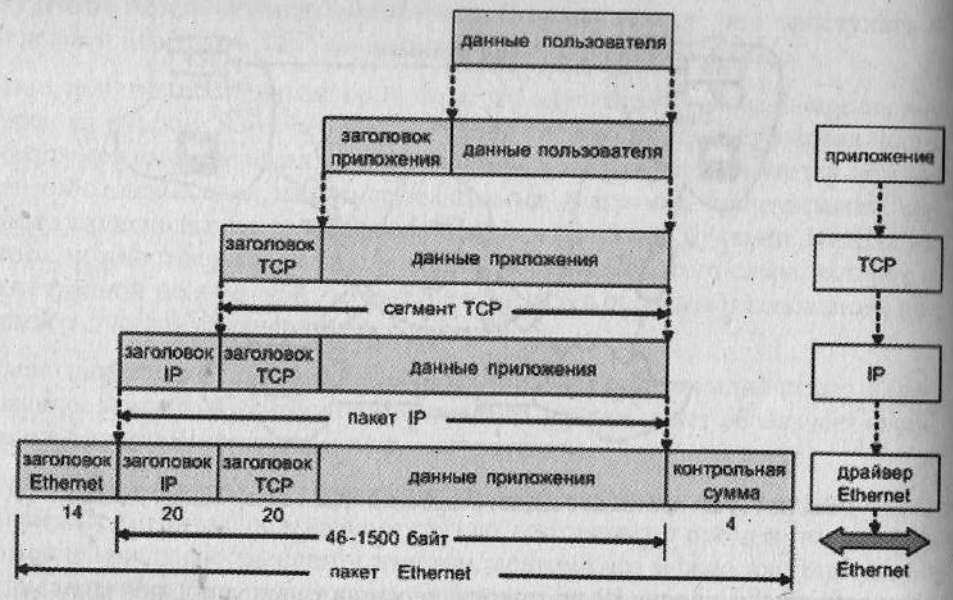


Рис. 12.2. Формирование пакетов при движении данных по стеку протоколов

Взаимодействие прикладных программ, использующих транспортные услуги протокола TCP, строится согласно модели «клиент-сервер», которая подразумевает, что одна программа (сервер) всегда пассивно ожидает обращения к ней другой программы (клиента). Для того чтобы клиент мог обращаться к необходимому ему сервису, установленному на сервере, он должен знать номер порта, по которому сервис ожидает обращения к нему.

Назначение номеров портов для программ, реализующих популярные общедоступные службы, осуществляется организацией **IANA** (Internet Assigned Numbers Authority), например, за службой удалённого доступа к файлам FTP закреплён номер 21, а за службой удалённого управления Telnet – номер 23. Во всех других случаях работающие программы вольны назначать произвольные номера портов. Обычно это происходит динамически, непосредственно перед обращением к сервису сервера, из списка свободных портов на данном узле.

Таким образом, номер порта совместно с номером сети и узла в этой сети однозначно определяют прикладной процесс, выполняющийся на конкретном компьютере сети. Этот набор параметров в англоязычной технической литературе получил название socket, что в переводе на русский язык означает – разъём. Отечественные специалисты не смогли смириться с этим переводом, который обозначает физическое разъёмное соединение, а подобрать

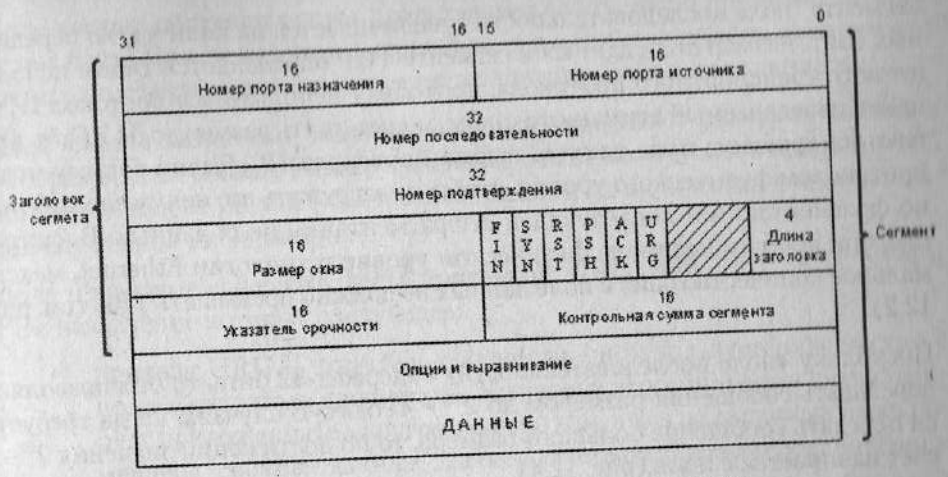


Рис. 12.3. Структура сегмента протокола TCP

подходящий синоним не захотели, поэтому, как и в большинстве аналогичных случаев, стали использовать слово socket в написании русскими буквами – сокет.

В обычной жизни совместную работу протоколов TCP и IP можно представить в виде совместной работы директора и его секретаря. Директор общается с другим директором на своём "логическом" уровне – на уровне названия компании и имени директора. Он пишет депешу другому директору, с просьбой срочно сообщить информацию по общему делу, обращаясь к нему по имени, с указанием компании, и передаёт секретарю. Секретарь помещает депешу директора в пакет (конверт) и снабжает его физическим адресом организации, а затем отправляет по ненадёжным каналам связи – по почте.

На приёмном конце, получив конверт в адрес организации, секретарь вскрывает конверт, определяет, что письмо адресовано директору, и перекладывает его в почту (папку входящие) директора. В результате один директор получил депешу от другого директора.

Если по истечении некоторого времени ответ не приходит, то директор компании, отправивший депешу, начинает процесс выяснения, почему нет ответа, потерялась ли депеша в пути и если да, то на каком этапе. В крайнем случае, инициируется процесс дублирования отправки. Таким образом, обеспечивается гарантированная доставка.

Прокол TCP всем байтам сообщения прикладного уровня присваивает номер, начиная с ноля. Поле "номер последовательности" содержит номер старшего байта в сегменте протокола TCP. Каждый раз при посылк

сегмента "поле последовательности" увеличивается на количество переданных байт. Размер поля данных в сегменте TCP определяется типом используемого транспортного протокола, поскольку используется протокол IP, то пакет, передаваемый этим протоколом, может иметь размер до 64 Кбайт. Однако на практике поле данных пакета протокола IP обычно согласуется с протоколом физического уровня, чтобы не загружать его ненужной работой по фрагментации и последующей дефрагментации поля данных. В частности, при использовании на физическом уровне технологии Ethernet, максимальное количество байт в поле данных не должно превышать 1460 (см. рис. 12.2).

Поскольку «поле последовательности» содержит 32 бита, то оно позволяет передавать сообщения размером до 2^{32} = 4Гбайт. В случаях, когда требуется передать сообщение большего размера, то по достижении значения $2^{32}-1$ счёт начинается с ноля (рис. 12.4).

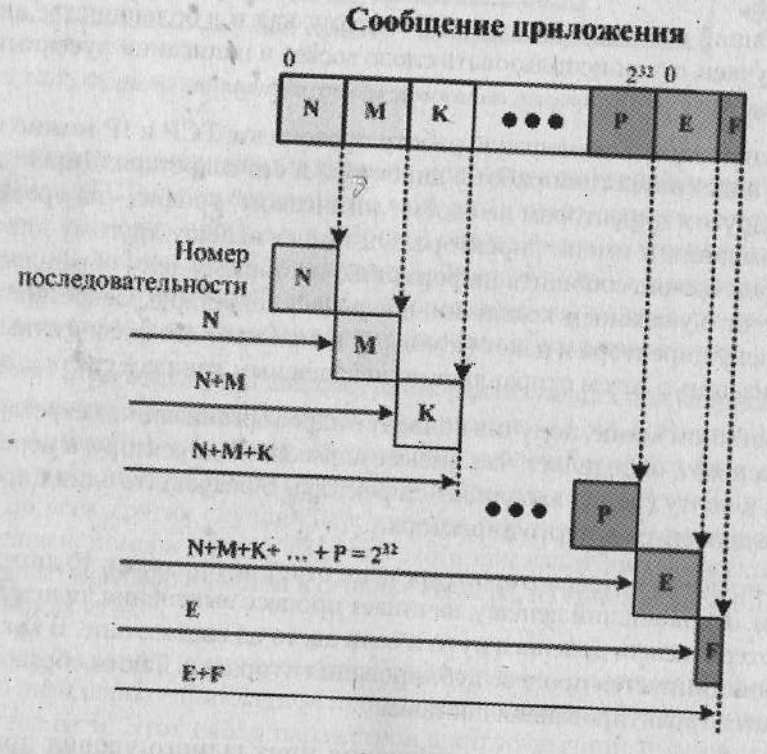


Рис. 12.4. Формирование значения поля «номер последовательности»

Поле «номер подтверждения» представляет собой число принятых получателем байт. При получении очередного сегмента получатель увеличивает «номер подтверждения» на величину количества байт в поле данных.

Поле «длина заголовка» занимает 4 бита и определяет размер заголовка в 32-х разрядных словах. Размер заголовка может колебаться в пределах от 5 до 15 слов или от 20 до 60 байт, в зависимости от количества дополнительной служебной информации.

Поле признаков содержит шесть признаков: URG, ASK, PSH, RST, SYN и FIN, назначение которых следующее:

- **признак URG** – указатель срочности, он используется совместно с полем "указатель срочности". Установка этого признака сигнализирует приложению на другом конце о том, что в поле данных были помещены "срочные данные", размер которых определяется значением 16-ти разрядного соответствующего поля;
- **признак ASK** – указатель того, что "поле подтверждения" используется. Установка этого признака регламентирует необходимость при обработке заголовка учитывать значение "поля подтверждения";
- **признак PSH**. Назначение этого признака заключается в том, что отправитель с его помощью оповещает получателя, что полученные данные необходимо как можно скорее передать приложению;
- **признак RST** – сброс. Установка этого признака приводит к экстренной остановке передачи данных и разрыву соединения;
- **признак SYN** – создание соединения. Установка этого признака обеспечивает начало соединения;
- **признак FIN** – окончание соединения. Установка этого признака говорит о том, что передающая сторона закончила передачу сообщений.

В одном сегменте могут присутствовать более чем один из перечисленных признаков, однако обычно устанавливается только один или два признака.

Поле «размер окна» определяет количество байт, которые получатель готов принять. Поскольку поле содержит 16 разрядов, то максимальный размер окна составляет 64 Кбайт. Прокол TCP, для обеспечения высокой надёжности доставки пакетов, использует метод скользящего окна с квитированием (см. главу 10).

Поле «указатель срочности» определяет размер срочных данных.

Поле «контрольная сумма сегмента» занимает 16 разрядов и рассчитывается

по всему сегменту и представляет собой сумму по модулю два (поразрядное сложение) всех 16-ти разрядных слов сегмента, кроме поля «контрольная сумма сегмента». Если рассчитанная контрольная сумма не совпадает со значением в поле «контрольная сумма сегмента» – сегмент будет уничтожен.

Контрольная сумма, помимо всего прочего, учитывает 96 бит (три 32-х разрядных слова) псевдозаголовка, который используется для внутреннего употребления и ставится перед заголовком TCP-сегмента, как показано на рис. 12.5. Этот псевдозаголовок содержит IP-адреса отправителя и получателя, название протокола и длину TCP-сегмента. Такой подход обеспечивает защиту от неправильных маршрутов сегментов, поскольку IP-адрес в совокупности с адресом порта однозначно определяют сокет соединения. Эта информация передается через интерфейс протокола TCP/IP в качестве аргументов для протокола IP.

Поле «опции и выравнивание»– необязательное поле и используется обычно для уточнения параметров сегмента. Механизм опций в базовом варианте представлен набором из 8-ми типов, однако по мере развития протокола TCP в него постоянно добавляются новые опции. Поскольку число опций может быть произвольным, а заголовок пакета должен содержать целое число 32-х слов, то для дополнения до 32-х разрядов недостающие разряды заполняются нолями.

Поле «данные» содержит данные, ответственность за которые несёт только отправитель. Протокол TCP, прежде чем приступить к отправке данных, сначала устанавливает соединение.

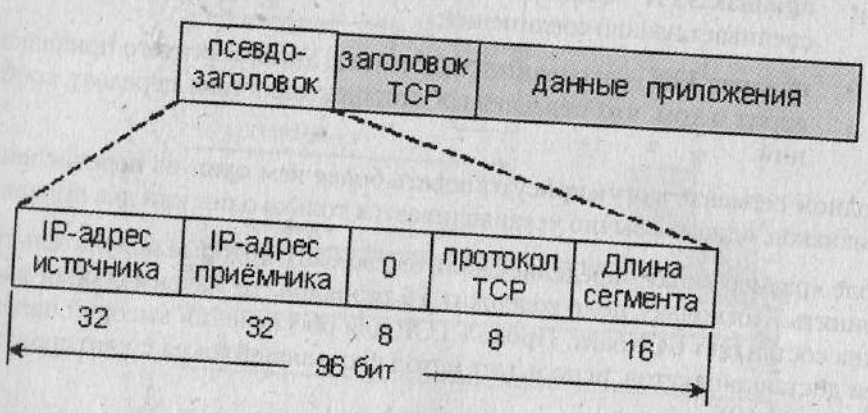


Рис. 12.5. Формат псевдозаголовка

12.1. Установка и разрыв соединения

Установка соединения начинается с посылки сегмента клиентом, в котором установлен признак SYN, указан номер порта, к которому клиент желает подключиться, свой номер порта и исходный номер последовательности клиента.

На данный сегмент сервер отвечает своим сегментом, в котором установлен признак SYN, указан исходный номер последовательности сервера и установлен признак ACK. Тем самым сервер подтверждает установление соединения и требует в последующем учёта значения «поля подтверждения».

Получив от сервера вышеописанный сегмент, клиент подтверждает получение сегмента отправкой собственного сегмента, в котором устанавливает признаки SYN и ACK, а в «поле подтверждения» помещает значение полученного «поля подтверждения» от сервера, увеличенное на единицу.

После этого считается, что соединение установлено. Каждая из сторон, отправляя свой сегмент с установленным признаком SYN, выбирает исходный номер последовательности для этого соединения и в поле опций устанавливает максимальный размер сегмента. Исходный номер последовательности меняется каждый раз, поэтому каждое соединение имеет свой отличный от других исходный номер последовательности. А указанный максимальный размер сегмента оговаривает максимально возможное количество байт в поле данных, всех последующих сегментов данного соединения.

Для закрытия соединения активная сторона должна послать сегмент с установленным признаком FIN. Принявшая сторона интерпретирует значение этого признака только как то, что в направлении с другой стороны движение данных прекращается. При этом сторона, получившая сегмент с признаком FIN, может посылать данные, однако на практике очень ограниченное количество приложений используют данную возможность.

Обычно клиент является инициатором, как открытия, так и закрытия соединения, это связано с тем, что клиентское приложение сначала требует некоторую информацию от сервера, а получив её, в услугах сервера перестаёт нуждаться.

Когда сервер получает сегмент с установленным признаком FIN, в зависимости от того, насколько быстро он может закрыть своё приложение и прекратить передачу, сервер ведёт себя по-разному.

Если сервер может сразу завершить приложение и передачу данных, то отправляет клиенту последовательно два сегмента, один сегмент с установленным признаком ACK, в поле «номер последовательности» которого помещает значение принятого поля последовательности, увеличенное на единицу, а второй – с установленным признаком FIN. После этого соединение

считается разорванным.

Если сервер сразу не может закрыть своё приложение и прекратить передачу данных, то он сначала отправляет сегмент, в котором устанавливает признак ACK, а в поле "номер последовательности" помещает значение принятого поля последовательности, увеличенное на единицу. Отправка данного сегмента означает, что сервер приступил к закрытию приложения. Затем он отправляет клиенту сегмент (ы) с данными.

После закрытия приложения сервер отправляет клиенту сегмент, подтверждающий закрытие и завершение передачи информации, в котором установлен признак FIN. Клиент обязательно подтверждает получение сегмента с признаком FIN, отправкой сегмента серверу с установленным признаком ACK, увеличив на единицу номер принятой последовательности. На этом процесс разрыва соединения заканчивается.

12.2. Выбор первоначального номера очереди

Протокол TCP не накладывает ограничения на многократное повторное использование конкретного соединения. Соединение задается посредством обмена трёх сегментов. Новые запросы на установление конкретного соединения будут рассматриваться протоколом как повторные реализации этого соединения. Вследствие чего возникает проблема: как отличить сегменты нового соединения от сегментов оставшихся от предыдущей реализации этого соединения? Данная проблема становится особенно актуальной, если соединение быстро открывается и закрывается несколько раз подряд или же если соединение прерывает свою работу с потерей информации, хранившейся в оперативной памяти компьютера, а затем устанавливается вновь. Вследствие чего необходимо иметь надёжный механизм, обеспечивающий исключение использования сегментов разных реализаций одного соединения с одинаковыми номерами очередей, даже если программа протокола TCP даст сбой и потеряет всю информацию об используемых ею номерах очередей.

В протоколе TCP при создании новых соединений в качестве первоначальных номеров очередей используется значение 32-х разрядного таймера, значение которого меняется каждые 4 микросекунды. Полный цикл таймера с шагом в 4 мкс составляет примерно 4,55 часа. Следовательно, начальные номера очередей будут повторяться с периодом примерно в 4,5 часа, а в пределах этого периода они будут уникальны.

В более поздних реализациях протокола TCP период изменения таймера первоначальных номеров увеличен с 4 до 8 мкс, что позволяет в течение

примерно 9,5 часов иметь уникальные начальные номера очередей.

Для каждого соединения существуют два номера очередей: один номер очереди отправителя, а другой – номер очереди получателя. Начальный номер очереди отправления определяется программой TCP, посылающей данные, а первоначальный номер очереди получателя определяется во время установления соединения.

Во время установления или инициализации какого-либо соединения обе программы отправителя и получателя протокола TCP должны синхронизировать друг с другом первоначальные номера очередей. Это осуществляется посредством обмена сегментами, устанавливающими соединения, содержащими признак SYN и исходные номера очередей.

Процесс синхронизации требует, чтобы каждая сторона, участвующая в соединении, посылала свой собственный первоначальный номер очереди, а также получала подтверждение на это от напарника. Кроме этого, каждая сторона должна также получить первоначальный номер очереди от напарника и послать подтверждение. Процесс синхронизации двух приложений A и B выглядит следующим образом:

1. A → B SYN: мой номер очереди X
2. A ← B ACK: ваш номер очереди X
3. A ← B SYN: мой номер очереди Y
4. A → B ACK: ваш номер очереди Y

Шаги 2 и 3 можно объединить в одно сообщение. Подтверждения номеров очередей необходимы, поскольку номера очередей не привязываются к глобальным часам данной компьютерной сети, и программы, реализующие протокол TCP у отправителя и получателя, могут иметь различные механизмы для выбора начальных номеров очередей. Кроме этого, получатель первого сегмента с признаком SYN не может знать, насколько задержался этот сегмент при передаче.

12.3. Таймеры протокола TCP

Протокол TCP – это надёжный протокол транспортного уровня. Высокая надёжность протокола обеспечивается тем, что каждый участник обмена подтверждает получение данных. Однако сегменты данных могут быть потеряны и подтверждения не получены. В таких случаях протокол TCP реализует механизм повторной передачи данных. Для реализации механизма повторной передачи протокол TCP ведёт четыре таймера для каждого соединения:

- таймер повторной передачи;
- таймер опроса размера окна передачи;
- таймер времени жизни соединения;
- таймер время жизни сегмента.

Таймер повторной передачи

Таймер повторной передачи предназначен для отсчёта времени, через которое протокол TCP должен произвести повторную передачу сегмента в случае отсутствия подтверждения. Все таймеры протокола TCP изменяют своё значение каждые 0,5 сек, однако прежде чем использовать таймер повторной передачи, следует определить время возврата подтверждения соответствующего сегмента.

Начальное время возврата измеряется при установлении соединения, когда на сегмент, содержащий признак SYN, посылается в ответ сегмент, содержащий признак АСК, и округляется в большую сторону. Однако полученное при первом измерении время возврата со временем может измениться, поскольку может измениться маршрут следования сегментов или загрузка сети. Соответственно протокол TCP должен отслеживать эти изменения и своевременно модифицировать значение таймера повторной передачи.

В любом случае измеренное время возврата в чистом виде не используется, оно увеличивается на коэффициент запаса. В разных реализациях протокола TCP коэффициент запаса составляет величину 2 или 4. Кроме этого, в большинстве случаев первый период повторной передачи принудительно устанавливается в 1,5 секунды. Если за это время сегмент подтверждения не пришёл, то для следующего сегмента значение таймера повторной передачи увеличивается вдвое, причём верхний предел ограничен 64 секундами. Таким образом, время повторной передачи может составлять 3, 6, 12, 24, 48 и 64 секунды.

В результате протокол TCP в худшем случае будет каждые 64 секунды повторять отправку сегмента, на который не было получено подтверждение. В большинстве реализаций протокола TCP максимальное время ожидания подтверждения принято равным 9 минутам. За это время отправитель 12 раз повторит отправку сегмента.

Таймер опроса размера окна передачи

Прежде чем приступить к отправке сегментов, отправитель должен получить от получателя сегмент, определяющий размер окна. Такое действие

необходимо, потому что получатель, принимая информацию, помещает её в свой буфер, размер которого определяется динамически приложением и с течением времени может меняться, тем самым приложение получателя регулирует темп поступления информации вплоть до полной остановки, если размер окна указывается равным нолю. Информация приложению передаётся только после заполнения буфера, поэтому пока буфер не будет заполнен, приложение не получит ни одного байта информации. Таким образом, размер окна определяется размером буфера получателя. Если размер окна превышает размер буфера, то после заполнения буфера вся последующая принимаемая информация будет утрачена.

После получения отправителем от получателя сегмента, содержащего информацию о размере окна, отправитель немедленно начинает посылать данные. К сожалению, в протоколе TCP не предусмотрена процедура подтверждения сегментов, содержащих служебную информацию, протокол TCP посылает подтверждения только на сегменты содержащие данные, поэтому возможна ситуация, когда сегмент с информацией о размере окна потерян.

Если сегмент с информацией об открытии окна потерян, то каждый участник соединения будет ждать действий от другого участника: получатель будет ожидать получение данных, поскольку он отправил отправителю размер окна, а отправитель надеется получить информацию о размере окна, которая позволит ему продолжить передачу.

Чтобы выйти из подобной тупиковой ситуации, отправитель должен осуществлять периодический опрос получателя на предмет, не был ли изменён размер окна. Период опроса определяет таймер опроса размера окна передачи, сегменты, которые при этом посылает отправитель, называются пробами окна.

Первый период передачи сегмента пробы окна принудительно устанавливается в 1,5 секунды. Если за это время сегмент с информацией об изменении окна не пришёл, то для следующего сегмента значение таймера опроса размера окна передачи увеличивается вдвое, причём верхний предел ограничен 60 секундами. Поскольку все таймеры протокола TCP работают с шагом 0,5 с, то период посылки проб окна может составлять 1.5, 3, 6, 12, 24, 48 и 60 секунд. Пробы окна будут отправляться с 60-секундным интервалом до тех пор, пока окно не будет открыто или приложение не решит, что соединение должно быть разорвано.

Таймер времени жизни соединения

Иногда возникает ситуация, когда приложения на обоих концах соединения запущены, но никакие данные в течение длительного времени не передаются. В

этом случае оба участника соединения должны знать, что произошло с другим участником соединения: работает ли на другом конце соединения приложение, или оно уже давно остановлено и нет нужды в поддержке данного соединения; или узел вышел из строя и был выключен; или узел вышел из строя и перезагрузился. При этом подразумевается, что ни одно из приложений – клиент или сервер – не имеет таймеров на прикладном уровне, которые позволяют определить отсутствие активности по соединению и прекратить работу приложения.

Для принудительного разрыва неактивного соединения и используется таймер времени жизни соединения. Таймер времени жизни соединения отсчитывает интервал равный 2-м часам. Его значение всегда сбрасывается в ноль при осуществлении любого обмена между приложениями. Таймер жизни соединения запускается на компьютерах каждого их участников соединения.

Если по истечении 2-х часов между участниками соединения не было никаких обменов, то участник, у которого первым сработал таймер жизни соединения, поскольку был запущен раньше, посылает второму участнику пробный сегмент «оставайся в живых». В случае получения подтверждения, первый участник устанавливает в ноль значение таймера времени жизни соединения и начинает отсчёт заново. Такая процедура будет продолжаться бесконечно каждые два часа. Здесь следует заметить, что некоторые реализации протокола требуют обязательного наличия данных в поле данных, в противном случае ответ на такой сегмент не посылается.

Если на пробный сегмент «оставайся в живых» ответа не последовало, то первый участник отправляет 10 пробных сегментов с интервалом 75 секунд, перед тем как прекратить попытки и разорвать соединение.

Таймер время жизни сегмента

Время жизни сегмента – это максимальное время, в течение которого сегмент может существовать в сети, перед тем как он может быть отброшен. Поскольку все сегменты отправляются с помощью протокола IP, а он имеет специальное поле – время жизни пакета, то «заблудившийся» пакет будет уничтожен средствами этого протокола. Поэтому время жизни сегмента – это несколько другое время.

Время жизни пакета в протоколе TCP применяется для случая разрыва соединения. Когда один из участников осуществляет активный разрыв соединения, то он в ответ на сегмент, содержащий признак FIN, посылает сегмент с признаком ACK. При этом соединение должно быть активным ещё в течение некоторого времени, достаточного для повторной посылки сегмента с признаком ACK, в случае утраты предыдущего сегмента. Кроме этого,

локальный номер порта клиента не может быть повторно использован, пока не будет разорвано соединение. Сервер же тоже не может использовать известные порты для нового соединения, пока не будет разорвано старое.

Время жизни сегмента играет ещё одну важную роль. Пока соединение полностью не разорвано и находится в стадии завершения, любой задержавшийся сегмент, прибывающий по соединению, отбрасывается, поскольку уже нельзя использовать данное соединение. Это делается для того, чтобы опоздавшие пакеты не были восприняты как часть нового соединения. Время жизни сегмента в разных реализациях протокола устанавливается в размере 1, 2 или 4 минуты.

В некоторых случаях возможна ситуация, когда сегмент с подтверждением на разрыв соединения потеряется и не придёт никогда. Поскольку приложение, которое осуществило активное закрытие, ожидает приёма данных, то в этом случае запускается таймер ожидания ответа. Если соединение не используется в течение 10 минут плюс 75 секунд, протокол TCP принудительно разрывает соединение.

12.4. Размер окна

Прежде чем начать передачу данных, оба участника соединения должны оговорить максимальный размер сегмента и размер окна. Для этой цели используется поле опций. Каждый раз при посылке сегмента с установленным признаком SYN клиент, инициирующий соединение, в поле опций определяет максимальный размер сегмента, которыми участники соединения будут обмениваться информацией. Под размер сегмента в поле опций отводится 16 разрядов, что позволяет определить максимальный размер сегмента в $2^{16} = 64$ Кбайт.

В заголовке сегмента протокола TCP (см. рис. 12.2) под размер окна отводится 16 разрядов, поэтому размер окна не может превышать $2^{16} = 64$ Кбайт. Однако, как было отмечено в предыдущих главах, смысл механизма скользящего окна сводится к тому, чтобы обеспечить максимальную скорость передачи данных по каналам связи, следовательно, в окно должно помещаться несколько сегментов. Для увеличения размера окна в протоколе TCP предусмотрена опция «коэффициент масштабирования» размером в 8 бит. Из возможных 15 комбинаций комбинация 15 не используется, поэтому наибольшее значение «коэффициента масштабирования» составляет – 14. Размер окна, с учётом «коэффициента масштабирования», определяется следующим образом:

$$W = W_0 \times 2^k$$

где:

- W_0 – размер окна, объявленный в заголовке сегмента TCP;
- k – значение "коэффициента масштабирования";
- W – размер окна.

Таким образом, максимальный размер окна может достигать:

$2^{16} \times 2^{14} = 2^{30}$ байт, что составляет 1 Гбайт.

После объявления размера окна и размера сегмента процесс передачи данных выглядит следующим образом. После прихода ответа на первый сегмент окно смещается вправо на одну позицию, как показано на рис. 12.6. Теперь отправитель может послать и сегмент 8, и так далее.

Отправитель может послать n сегментов окна, прежде чем будет получено подтверждение на очередной сегмент. В протоколе TCP используется метод "групповой квитанции", чтобы не загружать каналы связи. В большинстве реализаций протокола TCP получатель посылает одно подтверждение на два полученных сегмента, при этом для окна используется три указателя.

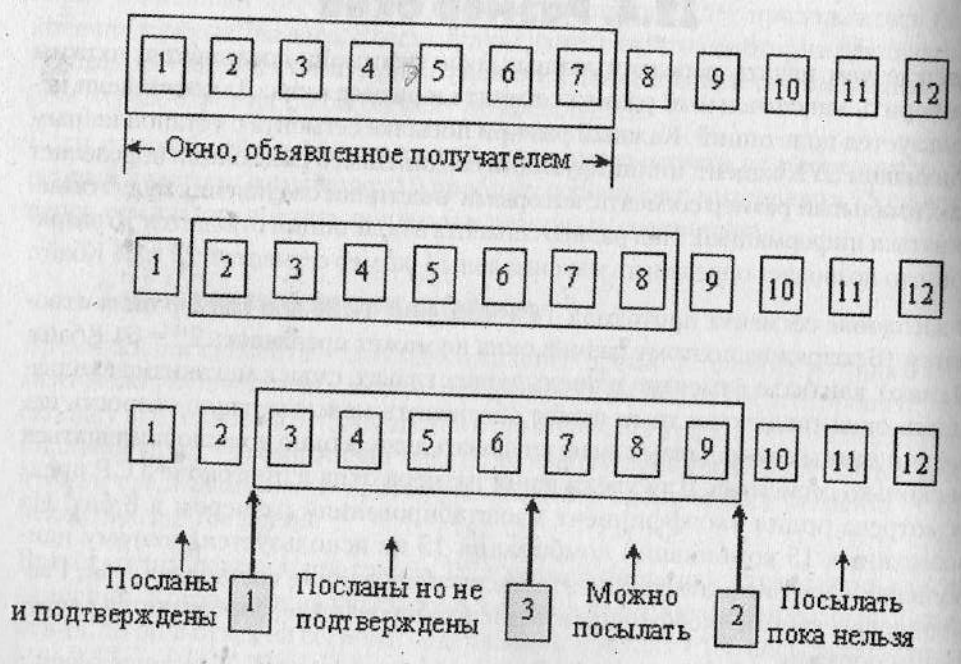


Рис. 12.6. Передача сегментов при использовании скользящего окна

Первый указатель определяет положение левого края окна, отделяя посланные сегменты, на которые получено подтверждение, от посланных сегментов, на которые подтверждение ещё не получено. Второй указатель отмечает правый край окна и указывает на сегменты, которые могут быть посланы до получения очередного подтверждения. Третий указатель отмечает границу внутри скользящего окна между уже посланными сегментами и теми, которые еще предстоит послать. Если указатель 3 совпадет с указателем 2, то отправитель прерывает дальнейшее отправление сегментов до получения хотя бы одного подтверждения.

Полученные данные от отправителя протокол TCP получателя передаёт порциями запросившему эти данные приложению. Размер порции определяется самим приложением, поэтому получаемые данные сначала помещаются в буфер, а затем, после его полного заполнения, передаются приложению. Обычно приложение достаточно быстро переписывает данные из буфера, однако иногда приложение может быть не готово принять очередную порцию информации. В этом случае возникает ситуация «переполнение буфера». При заполнении буфера все последующие поступившие сегменты будут просто отброшены.

Чтобы избежать лишних пересылок и ненужного увеличения нагрузки на сеть, вызванных заполнением буфера получателя, получатель должен остановить передачу данных отправителем. Механизм регулирования потока данных от отправителя достаточно прост. Получатель может послать отправителю сегмент, в котором указан изменённый, в сторону уменьшения, размер окна. Если указатели 2 и 3, изображённые на рис. 12.6, совпадут, то отправитель остановит передачу данных, поскольку все сегменты разрешённого окна переданы, а подтверждения ещё не получены.

Естественно, что получатель должен, не дожидаясь заполнения буфера, заранее изменить размер окна. Механизмы предсказания размера окна, когда в буфере остаётся мало свободного места, достаточно сложны, поскольку они должны учитывать размер буфера, размер сегмента, размер окна, скорость передачи данных по сети и вероятность освобождения буфера в ближайшее время приложением. Обычно механизм предсказания размера окна начинает работать при заполнении буфера на 60 – 80%.

Получатель может полностью остановить процесс передачи сегментов отправителем, послав ему сегмент с указанием размера окна равным нолю. Для того чтобы отправитель не начал процедуру разрыва соединения, вследствие длительного отсутствия связи с получателем, протокол TCP регламентирует необходимость в посылке получателем отправителю повторно сегментов с указанием размера окна равным нолю каждые 2 минуты.

Отправитель тоже имеет буфер, размер которого совпадает с размером окна. Поэтому отправитель сначала заполняет буфер, а затем начинает передачу данных. В зависимости от размеров буфера время на его заполнение может колебаться в достаточно широких пределах.

В некоторых случаях приложению получателя информация от отправителя может потребоваться срочно, для этого используется признак PUSH. Когда отправитель получает сегмент с установленным признаком PUSH, он должен немедленно, не дожидаясь заполнения своего буфера, приступить к отправке его содержимого полностью. Такая процедура в протоколе TCP получила название «проталкивания».

12.5. Протокол UDP

Протокол TCP является достаточно сложным протоколом и требует значительных временных ресурсов, как для установления соединения, так и для обеспечения гарантированной доставки. Поэтому скорость передачи данных по сети с использованием протокола TCP недостаточно высока.

В некоторых случаях потеря пакета может не оказывать существенного влияния на весь массив передаваемых данных. Например, при передаче видеоизображения потеря одного или даже нескольких пакетов не оказывает существенного влияния на качество получаемого видеоизображения. Однако скорость передачи, в данном случае, имеет существенное значение. Именно для такого рода приложений, в которых потеря информации не играет большой роли, а скорость передачи информации является определяющей, и был в 1980 году разработан «упрощённый» протокол передачи данных – UDP (User Datagram Protocol) – протокол дейтаграмм пользователя, который официально определён в документе RFC 768.

Протокол UDP является одним из двух основных протоколов транспортного уровня, расположенных непосредственно над IP-протоколом. Он предоставляет прикладным процессам транспортные услуги, однако обеспечивает ненадёжную доставку дейтаграмм. Другими словами, его пакеты могут быть потеряны, продублированы или прийти не в том порядке, в котором они были отправлены. Протокол UDP подразумевает, что проверка ошибок и исправление либо не требуются, либо должны выполняться приложением.

Протокол UDP имеет собственную систему адресации, его дейтаграмма имеет структуру, представленную на рис. 12.7.

Поскольку задачей транспортного уровня является передача информации между прикладными задачами, то дейтаграмма протокола UDP содержит

31 0

16 Порт отправителя	16 Порт получателя
16 Размер сообщения	16 Контрольная сумма
ДАННЫЕ	

Рис. 12.7. Структура дейтаграммы протокола UDP

два поля – порт получателя и порт отправителя. После того как дейтаграмма будет доставлена в компьютер-получатель средствами протокола IP, данные необходимо направить конкретному процессу-получателю.

Пакеты, поступающие на транспортный уровень, операционная система организует в виде очередей к точкам входа различных прикладных процессов. В терминологии протоколов TCP/IP такие системные очереди называются портами. Прикладной процесс, предоставляющий некоторые услуги другим прикладным процессам (сервер), ожидает поступления сообщений в порт, специально выделенный для этих услуг. Сообщения, отправляемые процессами-клиентами, должны содержать запросы на предоставление услуг. Порты нумеруются, начиная с нуля.

Как было отмечено ранее, назначение номеров портов прикладным процессам осуществляется либо централизованно организацией IANA, если эти процессы представляют собой популярные общедоступные сервисы, либо локально для тех сервисов, которые ещё не стали столь распространенными, чтобы за ними закреплять стандартные номера.

Локальное присвоение номера порта заключается в том, что разработчик некоторого приложения просто связывает с ним любой доступный, произвольно выбранный числовой идентификатор, обращая внимание на то, чтобы он не входил в число зарезервированных номеров портов. В дальнейшем все удалённые запросы к данному приложению от других приложений должны адресоваться с указанием назначенного ему номера порта.

Поле размер сообщения определяет общий размер дейтаграммы в байтах. Поскольку данное поле 16-ти разрядное, то дейтаграмма может иметь размер до 64 Кбайт (2^{16}).

Контрольная сумма вычисляется аналогично контрольной сумме протокола TCP и учитывает 96 бит (три 32-х разрядных слова) псевдозаголовка, как

показано на рис. 12.5. Этот псевдозаголовок содержит IP-адреса отправителя и получателя, название протокола и длину дейтаграммы. Данный подход обеспечивает защиту от неправильных маршрутов дейтаграмм, поскольку IP-адрес в совокупности с адресом порта однозначно определяют сокет соединения.

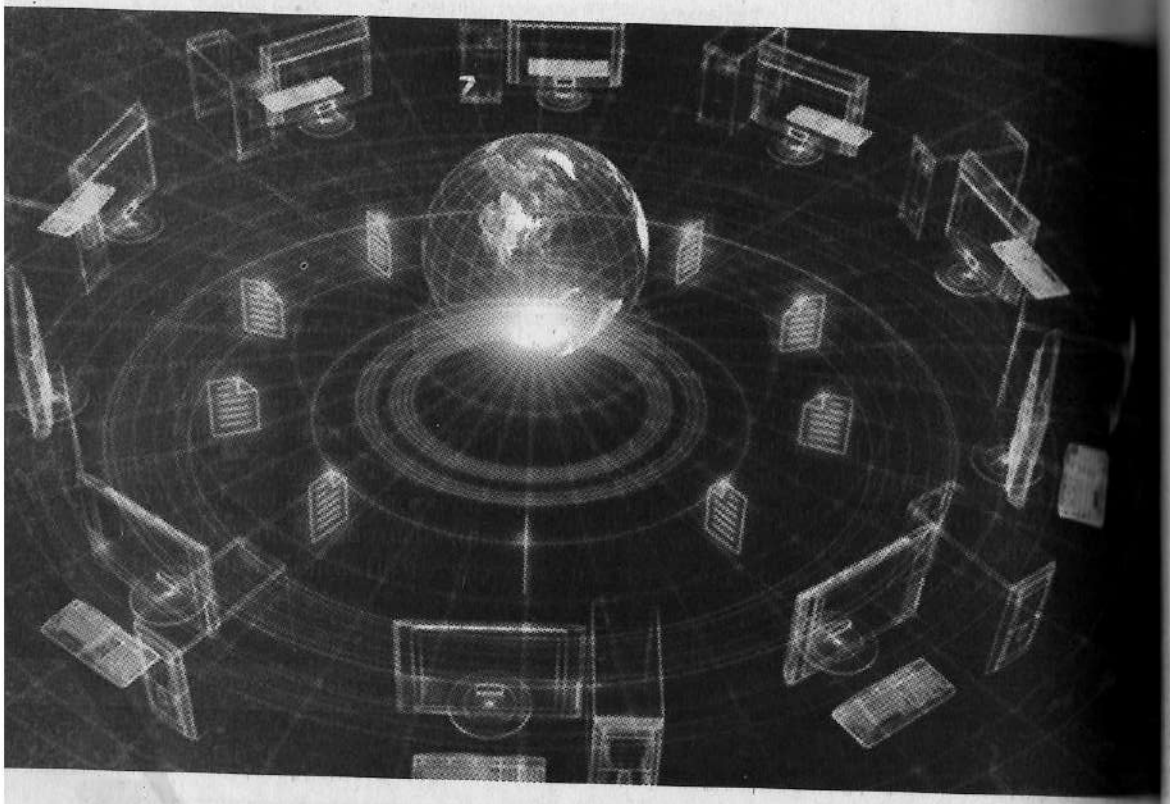
Контрольные вопросы

1. *Объясните назначение протокола TCP.*
2. *Каким образом определяется адрес точки входа процесса – сокет?*
3. *Что представляет собой порт назначения?*
4. *Кем определяются номера портов?*
5. *Зачем в TCP-протоколе используется псевдозаголовок?*
6. *Из каких основных частей состоит TCP-сегмент?*
7. *Какую структуру имеет заголовок TCP-сегмента?*
8. *Каково назначение поля "номер последовательности"?*
9. *Как определяется содержимое поля "номер последовательности"?*
10. *Объясните назначение поля "размер окна".*
11. *В чём заключается назначение поля опций?*
12. *Какие признаки присутствуют в заголовке TCP-сегмента?*
13. *Каково назначения поля "указатель срочности"?*
14. *Какую структуру имеет псевдозаголовок TCP-сегмента?*
15. *Как осуществляется установка соединения?*
16. *Как осуществляется разрыв соединения?*
17. *Как осуществляется выбор первоначального номера очереди?*
18. *Какие таймеры ведёт протокол TCP?*
19. *Объясните назначение таймера повторной передачи.*
20. *Зачем нужен таймер размера окна?*
21. *Каково назначение таймера времени жизни соединения?*
22. *Поясните назначение таймера жизни сегмента?*
23. *Какой размер может иметь окно в протоколе TCP?*

24. *Чем и как определяется размер окна?*
25. *Поясните термин "проталкивание".*
26. *Объясните назначение протокола UDP.*
27. *В каких случаях целесообразно использование протокола UDP?*

Глава 13.

ОТКАЗОУСТОЙЧИВЫЕ СЕТИ



Современные компьютерные сети могут содержать огромное количество сетевого оборудования – коммутаторы и маршрутизаторы разных уровней и степени сложности. Многие корпоративные пользователи переносят в свои IP-сети критичные к качеству и надёжности приложения, такие как телефония, видеоконференции, финансы, электронная коммерция и т.д. При этом любой отказ компьютерного оборудования и каналов связи может привести к выходу части или всей сети из строя. Поэтому проблема надёжности (отказоустойчивости) сети для многих владельцев сетей представляет собой первостепенную задачу, даже более важную, нежели производительность последней.

На отказоустойчивость сети влияет множество факторов, начиная от надёжности блока питания или порта отдельного коммутатора до ошибок в программном обеспечении маршрутизатора или узла сети. Поэтому проблема отказоустойчивости затрагивает большое количество аспектов сети и её условно можно разделить на три части: оборудование, каналы связи и программное обеспечение (ПО). Для создания высоконадёжной сети необходимо обеспечить надёжную работу каждой из частей.

Поскольку на качество программного обеспечения создатели сети не могут влиять напрямую – они могут только писать гневные письма в адрес разработчиков ПО или отказываться от него, поэтому следует сосредоточиться только на оборудовании и каналах связи.

Поскольку нельзя создать сеть, функционирующую без сбоев в течение очень длительного времени, то в теории надёжности используется показатель – вероятность безотказной работы в течение определённого периода времени.

Этот показатель для сверхнадёжных сетей составляет 99,999 %. Единственным способом повышения надёжности сетей является избыточность (резервирование) по каждому ключевому элементу.

В связи с тем, что каналы связи в настоящее время в основном представляют собой проводные, как оптические, так и медные кабели, которые обладают более высокой надёжностью по сравнению с любым оборудованием, то основное внимание следует уделять именно оборудованию. При этом не следует уж очень сильно пренебрегать возможностью резервирования наиболее важных каналов связи. Например, единственный кабель связи с удалённым подразделением, подвешенный на столбах освещения, может выйти из строя (быть оборванным) по самой невероятной причине. Его может оборвать упавший высотный строительный кран или упавшее дерево, автомобиль, потерявший управление, врезавшийся в столб и поваливший его, нерадивый

водитель, везущий негабаритный груз, обильный снегопад, шквалистый ветер и т.д. Если этот кабель проложен под землёй, то там его тоже подстерегает множество опасностей, например обрыв в результате копки экскаватором канавы при ликвидации аварии на водопроводе, внезапный провал грунта, воздействие влаги и т.д.

Ликвидация подобных аварий может продлиться несколько дней и сопряжена с существенными материальными затратами. В ряде случаев это смертельно опасно для организаций, поскольку чревато их разорением. В таких случаях следует использовать избыточные (резервные) каналы связи, территориально проложенные достаточно далеко друг от друга, например, по параллельным улицам, чтобы при выходе одного их них можно было быстро переключиться на другой. Некоторые не очень грамотные проектировщики располагают резервные кабели в непосредственной близости от основного кабеля, например, подвешивают на тех же столбах, что и основной кабель, или под землёй прокладывают два кабеля. Такое резервирование в перечисленных выше случаях не даёт никакого выигрыша, поскольку упавший строительный кран оборвёт все провода, а экскаватор выдернет из земли всё, что там находится.

Естественно, что для обеспечения высокой надёжности сети необходимо использовать очень надёжное оборудование. Высокая надёжность оборудования обеспечивается не столько выбором качественного оборудования ведущих производителей, сколько возможностью дублирования и горячей замены особо критичных к сбоям узлов оборудования, таких как блоков питания; управляющих модулей; коммутационной матрицы (шины); портов и т.д. Однако какое бы надёжное оборудование не было использовано, оно обязательно откажет, и дело состоит даже не в том, насколько надёжно данное оборудование, а в том, когда оно откажет. А как показывает практика – оно откажет в самый неподходящий момент. Поэтому применение даже сверхнадёжного оборудования не гарантирует от отказа сети.

Для ещё большего повышения надёжности оборудования следует использовать дублирование коммуникационного оборудования с применением избыточных каналов связи. При этом переход с основной линии связи на резервную или с основного устройства на резервное может происходить как в автоматическом режиме, так и вручную, при участии администратора. Очевидно, что автоматический переход повышает надёжность системы, так как время простоя сети в этом случае будет существенно меньше, чем при вмешательстве администратора. Для выполнения автоматических процедур реконфигурации необходимо иметь в сети интеллектуальные коммуникационные устройства, а также централизованную систему управления, помогающую устройствам распознавать отказы в сети и адекватно на них реагировать.

На уровне коммутаторов повышение надёжности сети может быть достигнуто за счёт применения коммутаторов второго уровня, поддерживающих алгоритм связующего дерева (см. главы 7 и 8) и агрегирования каналов. Справедливости ради следует отметить, что использование протоколов, поддерживающих алгоритм связующего дерева, хотя и обеспечивает переключение на резервные каналы связи, однако все узлы, подключённые к вышедшему из строя коммутатору, оказываются лишёнными доступа к сети на время его замены. Тем не менее, такой подход существенно повышает надёжность всей сети.

Алгоритмы агрегирования канала обеспечивают работоспособность сети при выходе канала связи из строя, поскольку используют несколько параллельно работающих каналов связи.

Наибольшие неприятности доставляет отказ маршрутизатора, поскольку отправляемая компьютером в Internet или другие сети информация, как правило, поступает на локальный маршрутизатор или коммутатор третьего уровня. Маршрутизатор или коммутатор передают эту информацию на другой (межсетевой) маршрутизатор, после чего данные попадают в Internet и другие сети. Обычно локальный маршрутизатор выступает для компьютеров данной подсети в роли шлюза. Если в подсети определён единственный маршрутизатор, то в случае выхода его из строя компьютеры теряют связь с другими подсетями и с Internet. Для обеспечения отказоустойчивости необходимо включить в каждую подсеть ещё один или несколько дублирующих шлюзов. При отказе основного маршрутизатора компьютер должен иметь возможность переключиться на резервный маршрутизатор. Механизм обнаружения неработающего шлюза и переключения на другой шлюз определён в документе RFC 816.

Для этих целей разработано несколько протоколов, например протоколы RIP и IRDP. Данные протоколы основаны на периодической рассылке групповых сообщений, в которых указывается приоритет маршрутизатора и время жизни сообщения.

Пример работы протоколов приведён на рис. 13.1. Допустим, компьютер желает обратиться к серверу, расположенному в сети университетского городка. На этом компьютере маршрутизатор А определён как маршрутизатор по умолчанию (основной), собственно через него и происходит общение данного компьютера с сервером. В один прекрасный момент маршрутизатор А выходит из строя.

Поскольку оба маршрутизатора обмениваются между собой «приветственной» информацией, то через некоторое время маршрутизатор В узнаёт, что маршрутизатор А не подаёт признаков жизни. Тогда маршрутизатор В

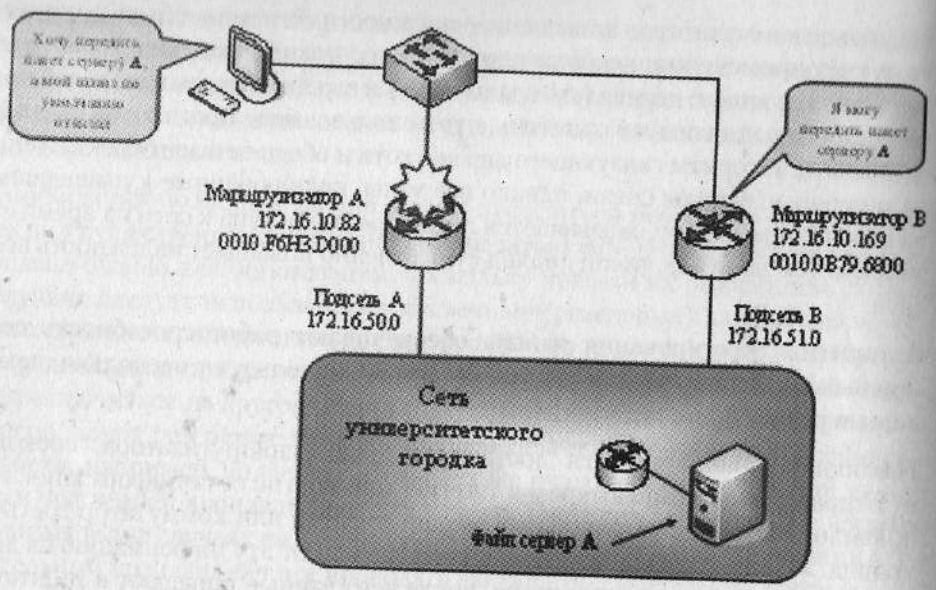


Рис. 13.1. Работа маршрутизаторов в сети при отказе одного из них

берёт на себя функции маршрутизатора А, при этом он всем узлам сети, подключённым к маршрутизатору А, в данном случае через коммутатор, посылает сообщение со своим IP-адресом с предложением о возможности передачи информации к серверу через себя. Теперь наш компьютер может организовать обмен информацией с сервером через маршрутизатор В. Правда, для этого на компьютере тоже должен быть установлен соответствующий протокол.

Несомненным преимуществом протоколов RIP и IRDP является возможность организации нескольких параллельных каналов, как показано на рис. 13.2. Однако протокол RIP для передачи информации по альтернативным каналам использует метрику – количество переходов до маршрутизатора. Поскольку на рис. 13.2 все маршрутизаторы имеют одинаковую метрику, то протокол RIP последовательно по каждому каналу будет передавать пакеты, при этом скорость каналов не учитывается. Поэтому для обеспечения быстрой передачи информации не рекомендуется использование каналов связи, имеющих большие различия в скорости передачи.

Недостаток протоколов RIP и IRDP заключается в том, что они медленно адаптируется к изменениям топологии сети. Их выполнение на всех узлах сети может оказаться неприемлемым по многим причинам, включая

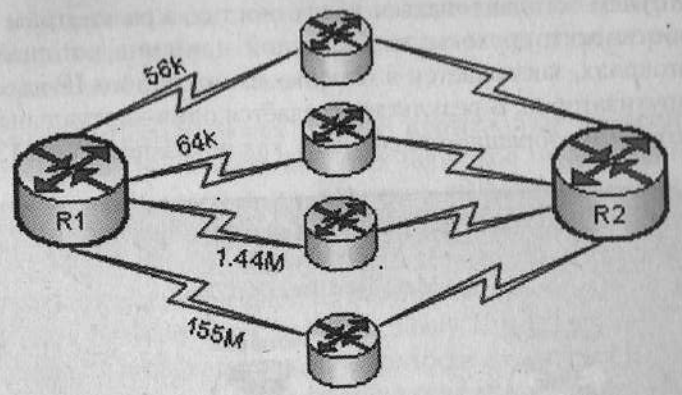


Рис. 13.2. Работа протокола RIP по нескольким каналам

административные издержки, издержки обработки данных, проблемы безопасности или отсутствие поддержки протоколов на некоторых платформах.

В конце 90-х годов, для уменьшения времени поиска и переключения на резервный маршрутизатор, созданное в 1986 году открытое международное сообщество проектировщиков, учёных, сетевых операторов и провайдеров – IETF (Internet Engineering Task Force) приступило к созданию протокола для резервирования маршрутизаторов. Данный протокол получил название – VRRP (Virtual Router Redundancy Protocol) – сетевой протокол, предназначенный для увеличения доступности маршрутизаторов, выполняющих роль **шлюза по умолчанию**. Однако в 1997 году компания Cisco сообщила, что подобный протокол HSRP (Hot Standby Redundancy Protocol) – протокол резервирования с горячей избыточностью – уже запатентован ею, тем не менее, сообщество IETF продолжило работу над протоколом VRRP. Поскольку протокол VRRP решил проблемы протокола HSRP, то фирма Cisco стала использовать протокол VRRP, утверждая, что он является их собственностью.

Из-за патента на протокол HSRP было невозможно создать свободно распространяемую версию протокола VRRP, поэтому программисты команды OpenBSD приступили к разработке альтернативного протокола CARP (Common Address Redundancy Protocol) – протокола избыточности общего адреса. Разработка протокола была завершена в октябре 2003 года, и первой операционной системой с поддержкой протокола CARP, естественно, стала OpenBSD. Позднее поддержка протокола появилась во FreeBSDи NetBSD, а затем и в DragonFlyBSD.

Поскольку все эти протоколы HSRP, VRRP и CARP построены на общих принципах и отличаются только частностями реализации и терминологией,

то мы не будем останавливаться на частностях, а рассмотрим общие принципы работы этих протоколов. Основной принцип, который заложен во всех протоколах, заключается в использовании одного IP-адреса для группы маршрутизаторов. В результате создаётся один виртуальный маршрутизатор, к которому обращаются все узлы, как показано на рис. 13.3.

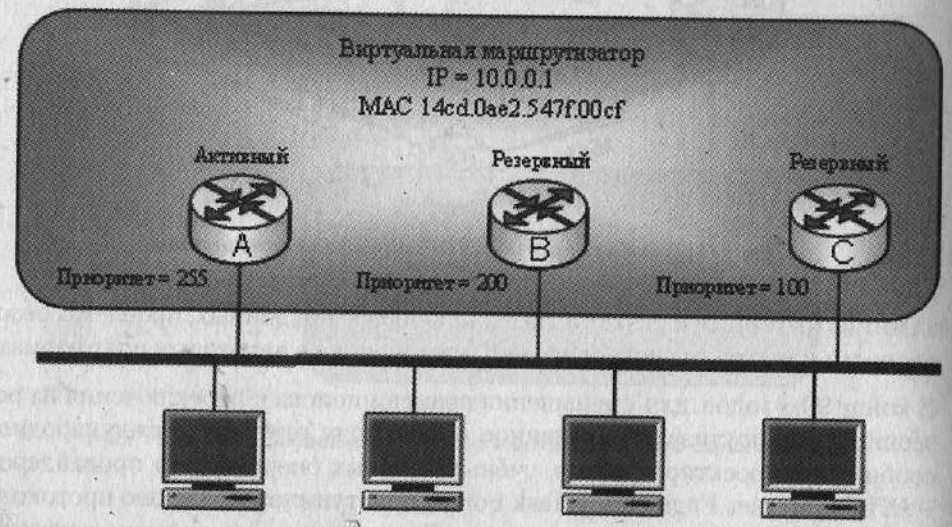


Рис. 13.3. Организация виртуального маршрутизатора протоколами HSRP, VRRP и CARP

Виртуальному маршрутизатору (в нашем примере состоящему из трёх маршрутизаторов) присваиваются виртуальные IP-адрес 10.0.0.1 и MAC-адрес 14cd.0ae2.544f.00cf, для каждого маршрутизатора определяется уровень приоритета. Далее все маршрутизаторы обмениваются служебной информацией, используя протокол UDP. Протокол UDP выбран только потому, что не требует для установления связи "рукопожатия" и поэтому работает быстрее протокола TCP, правда, при этом он не гарантирует доставку пакетов получателю – они могут потеряться. В рамках обмена служебной информацией используются пакеты приветствия (hello), которые отправляются на групповой адрес 224.0.0.x (в каждом протоколе используется свой групповой адрес, например, протокол HSRP использует групповой адрес 224.0.0.2, а протокол VRRP – адрес 224.0.0.18). Для передачи информации на канальном уровне – определяется групповой (виртуальный) MAC-адрес.

Маршрутизаторы обмениваются приветствиями, в которых указывают уровень своего приоритета, маршрутизатор, у которого уровень приоритета окажется выше, и становится основным или активным маршрутизатором.

Остальные маршрутизаторы становятся не активными или резервными. Все пакеты от узлов, подключённых к виртуальному маршрутизатору, обрабатывает только активный маршрутизатор.

Активный маршрутизатор через равные промежутки времени посылает пакеты приветствия, если в течение трёх периодов от активного маршрутизатора не поступает пакетов приветствия, то резервные маршрутизаторы расценивают такую ситуацию как выход из строя активного маршрутизатора. Пауза в три сообщения выбрана для того, чтобы наверняка удостовериться, что маршрутизатор вышел из строя, а не произошла потеря пакета, так как протокол UDP не гарантирует доставку. После чего они выясняют, у кого из оставшихся резервных маршрутизаторов уровень приоритета выше. Маршрутизатор с наивысшим уровнем приоритета становится активным, а остальные – резервными.

Среднее время перехода резервного маршрутизатора в активное состояние в разных протоколах разное и лежит в пределах от нескольких секунд до нескольких десятков минут. В любом случае, виртуальный маршрутизатор автоматически перестраивается и сохраняет работоспособность. А все узлы как посылали свои пакеты на виртуальные IP- и MAC-адреса, так и продолжают это делать, они ощущают только некоторую задержку в передаче, на время перестройки виртуального маршрутизатора.

Использование виртуального маршрутизатора резко повышает надёжность сети и сокращает время на восстановление связи при отказе активного маршрутизатора. Однако использование одного или нескольких дорогостоящих маршрутизаторов в горячем резерве приводит к крайне не эффективному использованию средств. Поэтому в последних версиях протоколов HSRP, VRRP и CARP введена возможность использования маршрутизаторов одновременно в двух ипостасях – в качестве активного и пассивного для разных групп.

На рис. 13.4 все узлы локальной сети объединены в две группы, связь которых с внешней сетью осуществляется через маршрутизаторы A и B, причём маршрутизатор A является активным для группы 1 и резервным для группы 2, а маршрутизатор B является активным для группы 2 и резервным для группы 1 соответственно. Таким образом, узлы каждой группы передают информацию во внешнюю сеть через отдельный маршрутизатор, что существенно повышает производительность. В то же время при отказе одного из маршрутизаторов его функции на себя берёт оставшийся "в живых" маршрутизатор. В результате решаются две проблемы – повышается и производительность, и надёжность.

В некоторых случаях поток информации между сетями может быть настолько большим, что с ним не может справиться ни один существующий даже самый мощный маршрутизатор неимоверной цены. Единственным выходом из создавшейся ситуации может быть только разделение пользователей на группы и применение виртуального маршрутизатора, содержащего маршрутизаторы двойного назначения – активный/резервный, как показано на рис. 13.4. Разделение пользователей на группы всегда происходит весьма болезненно, поскольку, с одной стороны, весьма сложно выработать критерии объединения, а с другой стороны, процесс разделения пользователей на группы в большинстве случаев сопровождается крайне негативной реакцией со стороны последних.

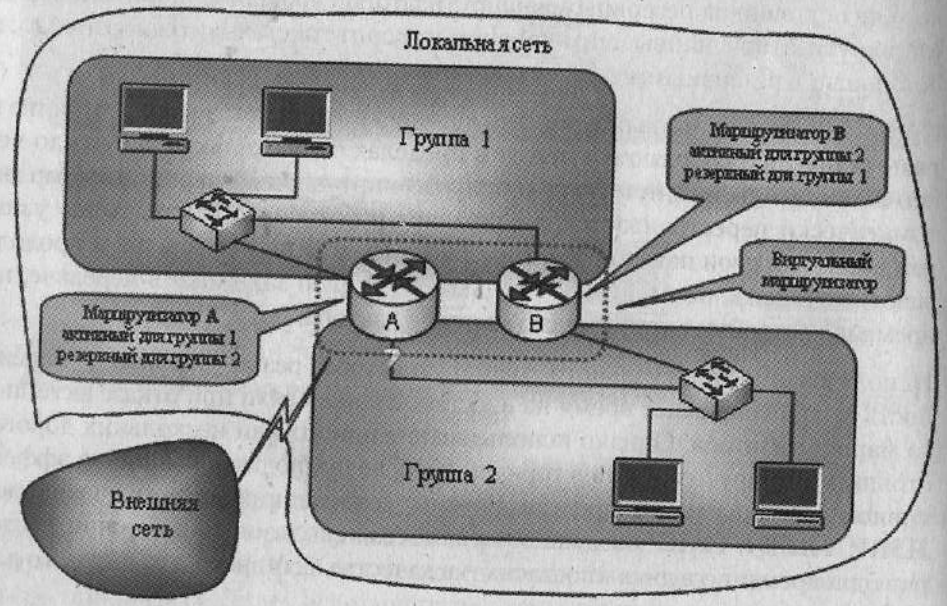


Рис. 13.4. Организация виртуального маршрутизатора протоколами HSRP, VRRP и CARP

Для решения этих не простых и животрепещущих проблем фирма Cisco разработала протокол GLBP (Gateway Load Balancing Protocol) – протокол балансировки нагрузки между маршрутизаторами, выполняющими роль шлюза по умолчанию. Протокол GLBP работает аналогично протоколам HSRP, VRRP и CARP, но не совсем. Если протоколы HSRP, VRRP и CARP позволяют объединять группу маршрутизаторов в один виртуальный маршрутизатор с общими IP- и MAC-адресами, в которой один маршрутизатор выбирается активным, а все остальные маршрутизаторы остаются резервными до тех пор, пока не произойдёт выход из строя активного маршрутизатора, обладая ресурсами, которые почти не используются во

время эксплуатации этой системы, то протокол GLBP обеспечивает распределение нагрузки на все маршрутизаторы, входящие в группу виртуального маршрутизатора, используя один виртуальный IP-адрес и несколько виртуальных MAC-адресов. В результате все маршрутизаторы, входящие в виртуальный маршрутизатор, участвуют в передаче пакетов.

В составе виртуального маршрутизатора протокол GLBP определяет один активный виртуальный маршрутизатор – AVG (active virtual gateway) и несколько активных виртуальных транспортёров AVF (active virtual forwarders), каждому из которых администратор присваивает виртуальный MAC-адрес. Выбор активного маршрутизатора AVG может осуществляться на основе приоритетов, задаваемых администратором, или на основе MAC-адреса маршрутизаторов.

За каждым маршрутизатором закрепляется индивидуальная группа узлов, которые будут работать только с этим маршрутизатором, таким образом, осуществляется распределение нагрузки. Каждый маршрутизатор всем узлам своей группы сообщает общий IP-адрес (IP-адрес виртуального маршрутизатора) и свой индивидуальный виртуальный MAC-адрес, как показано на рис. 13.5, на основании которых узлы устанавливают шлюз по умолчанию.

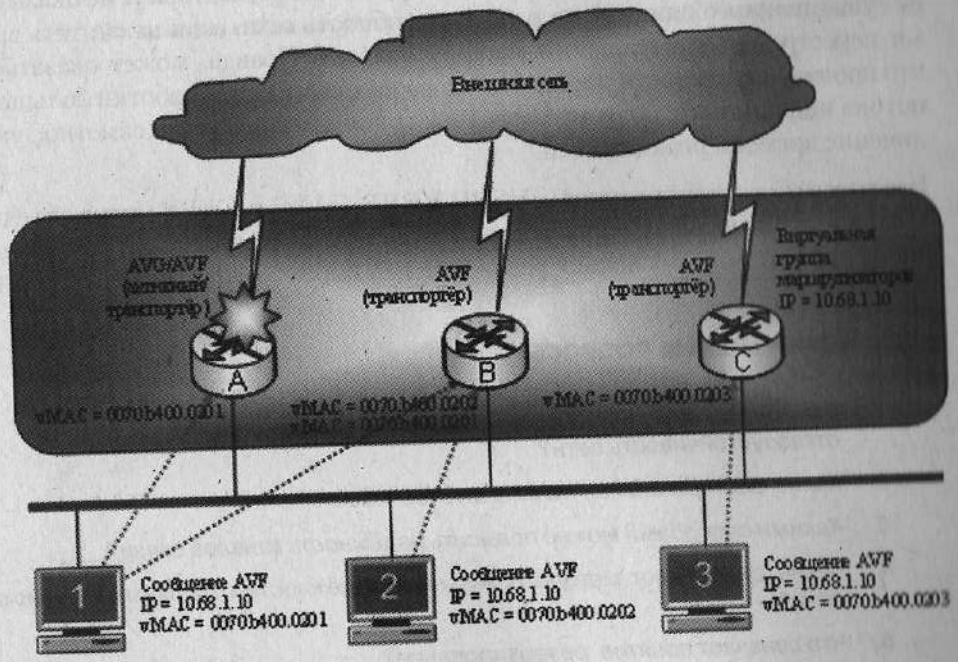


Рис. 13.5. Организация виртуального маршрутизатора протоколом GLBP

В примере, приведённом на рис. 13.5, используются три маршрутизатора A, B и C, имеющих виртуальный IP-адрес 10.68.1.10. Каждому маршрутизатору присвоены виртуальные MAC-адреса: маршрутизатору A – 0070.b400.0201, маршрутизатору B – 0070.b400.0202 и маршрутизатору C – 0070.b400.0203 соответственно. Активным маршрутизатором определён маршрутизатор A с MAC-адресом 0070.b400.0201. Три группы узлов настроены на соответствующий маршрутизатор и только через него имеют доступ к внешней сети.

По умолчанию каждые 3 секунды активный маршрутизатор A рассылает сообщения приветствия маршрутизаторам-транспортёрам B и C, которыми подтверждает свою работоспособность. Если в течение 10 секунд от активного маршрутизатора A не приходит ни одного сообщения приветствия, то транспортёры B и C расценивают такую ситуацию как выход из строя активного маршрутизатора A, после чего начинают процедуру выбора активного маршрутизатора. Допустим, после выборов активным становится маршрутизатор B.

Активный маршрутизатор B продолжает обрабатывать все пакеты, адресованные ему с MAC-адресом 0070.b400.0202, но кроме этого, он начинает обрабатывать и все пакеты адресованные маршрутизатору A, с MAC-адресом 0070.b400.0201. В результате выход из строя маршрутизатора A не оказывает существенного влияния на работоспособность сети, если не считать время перестройки виртуального маршрутизатора. Правда, может оказаться, что производительности маршрутизатора не хватит для обработки большего потока информации, поэтому узлы 1 и 2 групп могут ощутить заметное увеличение времени реакции сети.

Все рассмотренные протоколы HSRP, VRRP, CARP и GLBP могут объединять в виртуальный маршрутизатор не более 4-х маршрутизаторов. Этого вполне достаточно, поскольку реализуется четырёхкратное резервирование.

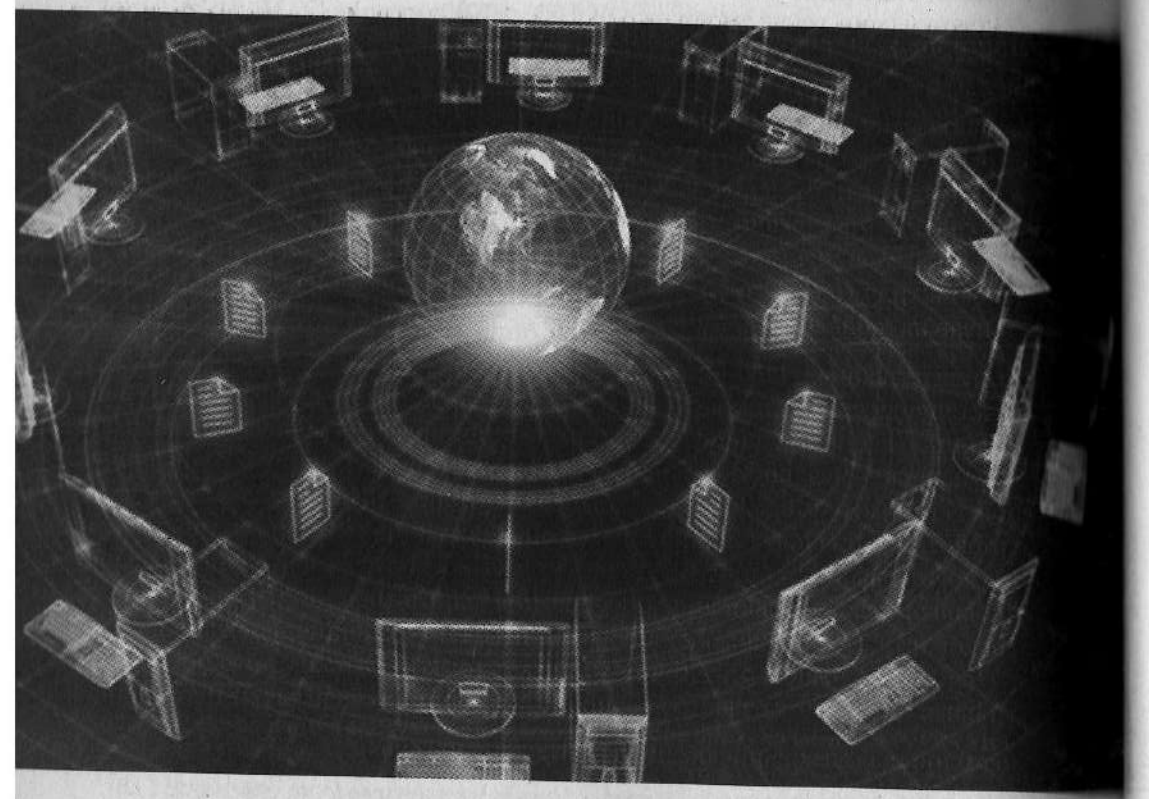
Контрольные вопросы

1. *Объясните, почему необходимо обращать внимание на отказоустойчивость сети?*
2. *Какие факторы влияют на отказоустойчивость сети?*
3. *Какими способами можно повысить надёжность каналов связи?*
4. *Какие существуют методы повышения надёжности отдельных устройств сети?*
5. *Что означает понятие резервирование?*
6. *Возможно ли повышение надёжности сети на уровне коммутаторов?*

7. *Какими способами можно добиться повышения надёжности сети на уровне коммутаторов?*
8. *Какими недостатками обладают протоколы RIP и IRDP?*
9. *Объясните, как работают протоколы RIP и IRDP.*
10. *Назовите назначение протокола VRRP.*
11. *Назовите назначение протокола HSRP.*
12. *Назовите назначение протокола CARP.*
13. *Объясните, как работают протоколы HSRP, VRRP и CARP.*
14. *Назовите основные отличия протоколов HSRP, VRRP и CARP.*
15. *В чём заключается принцип работы протокола VRRP?*
16. *В чём заключается принцип работы протокола HSRP?*
17. *В чём заключается принцип работы протокола CARP?*
18. *Возможно ли распределение нагрузки при использовании протоколов HSRP, VRRP и CARP?*
19. *В чём отличие протокола GLBP от протоколов HSRP, VRRP и CARP?*
20. *Объясните, как работает протокол GLBP.*
21. *Объясните, как происходит распределение нагрузки в протоколе GLBP.*

Глава 14.

БЕСПРОВОДНЫЕ СЕТИ



Использование кабельных систем передачи информации экономически целесообразно только в достаточно населённых пунктах – городах, посёлках и т.д., поскольку прокладка кабеля и установка соответствующего сетевого оборудования обходится довольно дорого. На территориях с низкой плотностью населения и для связи с удалёнными объектами экономически более выгодно использование беспроводных каналов связи.

Первые опыты по созданию беспроводных линий связи были проведены в СССР ещё в начале 30-х годов XX века. Тогда советские инженеры М. И. Греков и В. М. Большеверов провели опыты по созданию направленной радиосвязи между Москвой и Люберцами в дециметровом диапазоне. Первая же коммерческая линия радиосвязи была построена в США между Нью-Йорком и Филадельфией на ультракоротких волнах в 1935 году протяжённостью 150 км. Для перекрытия такого расстояния через каждые 50 км были установлены две промежуточные ретрансляторное станции – «релейные станции» в прямой видимости, которые принимали ослабленный сигнал, восстанавливали, усиливали его и передавали дальше. Сама радиолиния получила название – радиорелейной.

Первая радиорелейная линия в СССР была построена между Москвой и Рязанью в 1953 году. А в 1958 году, при создании опытной системы противоракетной обороны (системы «А») на полигоне Сары-Шаган в пустыне Бет-Пак-Дала в Казахстане, для передачи информации с локаторов точного наведения в центр обработки, расположенный недалеко от города Приозёрска на берегу озера Балхаш, были построены несколько радиорелейных линий с дальностью передачи от 150 до 200 км и скоростью передачи 1 Мбит/с.

В СССР была создана аппаратура магистральных и зоновых радиорелейных линий прямой видимости, работающих в диапазонах частот 2, 4, 6 и 8 ГГц, что позволило к концу 70-х годов построить радиорелейные линии общей протяженностью свыше 300 тыс. км. Они обеспечивали подачу программ центрального телевидения и звукового вещания на вещательные станции, а также передачу сигналов многоканальной телефонии.

В середине прошлого века был открыт новый механизм распространения дециметровых волн на дальние расстояния – тропосферное рассеивание. Суть явления заключается в свойстве отражения радиоволн от верхних слоев атмосферы (на высоте 10 – 12 км). Исследования, проведённые в США, Англии и СССР, показали возможность создания многоканальных тропосферных радиорелейных линий, в которых расстояние между соседними станциями может достигать 400 – 800 км при мощности передатчика 10 КВт. Для экономически и стратегически важных районов Сибири, Дальнего Востока

и Севера, с весьма низкой плотностью населения и слабым развитием всех видов связи, строительство тропосферных радиорелейных линий было чрезвычайно важным и перспективным.

Протяженность одной только тропосферной радиорелейной линии «Север» равнялась примерно 14 тысячам километров. Она состояла из 55 тропосферных радиорелейных станций, расположенных на расстоянии от 250 до 300 км друг от друга, и была развёрнута в конце 60-х годов прошлого века. В ней использовалась 60-ти канальная аппаратура, работающая в диапазоне 2 ГГц. Впоследствии переход на диапазон 5 ГГц позволил передавать сигнал чёрно-белого телевидения. На станциях устанавливались антенны размером 20x20 или 30x30 метров, по две на каждое направление. На рис. 14.1 приведена фотография одной из промежуточных станций на три направления. Зеркала антенн выполнены в виде части параболоида вращения, приподнятого над землёй, с рупорным облучателем на башенках. Из фотографии видно, что антенны направлены вверх, для приёма сигнала отражённого от тропосферы и для передачи сигнала в тропосферу.

Апогеем в развитии тропосферных радиорелейных линий связи стало создание в 1981 году сверхдальней линии связи Москва-Дели между городами Душанбе и Сринагар, расстояние между станциями превышало 700 км.

В настоящее время новейшие отечественные разработки, при массе

Рис. 14.1. Фото радиорелейной станции тропосферной связи «Север»

оборудования в 80 Кг, позволяют передавать информацию в цифровом виде со скоростью до 2 Мбит/с, что позволяет использовать их для оперативного развертывания линий связи в полевых условиях на расстояниях от 200 до 400 км от зоновых станций.

Успехи микроэлектроники позволили по-новому взглянуть на проблему беспроводной связи, особенно на небольшие расстояния в густонаселённых районах. В результате развитие систем беспроводной связи пошло по двум независимым направлениям: беспроводных сетей передачи данных и сетей мобильной связи.

14.1. Беспроводные сети передачи данных

В настоящее время общепринятым считается разделение беспроводных сетей передачи данных на три группы, различающихся дальностью передачи информации:

- PAN – персональные сети, это небольшие сети с радиусом до 10 м. Они предназначены для связи мобильных телефонов, принтеров, компьютеров и т. п. С помощью таких сетей реализуется простая синхронизация данных, простой обмен информацией в небольших рабочих группах. Наиболее распространённый стандарт для персональных сетей – Bluetooth;
- WLAN – локальные беспроводные сети с радиусом до 100 м. С их помощью реализуется беспроводной доступ к групповым ресурсам в здании, университетском городке и т. п. Обычно такие сети используются для продолжения проводных корпоративных локальных сетей. Основным стандартом для локальных беспроводных сетей является IEEE 802.11 (Wi-Fi);
- WWAN – беспроводные сети масштаба города, обеспечивают мобильным пользователям доступ к их корпоративным сетям и Интернету. Основным стандартом для беспроводных сетей масштаба города является IEEE 802.16 (WiMAX).

Общие принципы передачи информации, используемые в каждой из групп, очень похожи, однако имеют существенные отличия, поэтому рассмотрим каждую из них более подробно.

14.1.1. Персональные беспроводные сети – Bluetooth

Слово Bluetooth является переводом на английский язык датского слова Blatand, означающего «синезубый». Так прозвали одного из королей Дании,

который объединил враждовавшие датские племена в одно королевство. По аналогии с ним, Bluetooth призван объединять пользователей в единую сеть. Основное внимание при разработке спецификации для Bluetooth было уделено устройствам с низкой мощностью передатчика (до 100 мВт), поэтому только в 2002 году она стала частью стандарта IEEE 802.15.1.

Радиосвязь в стандарте Bluetooth осуществляется по принципу точка-точка в свободном от лицензирования диапазоне частот – 2,4 ГГц. В данном диапазоне выделяется 79 рабочих частот шириной в 1 МГц. Несущая частота скачкообразно меняется 1600 раз в секунду. Последовательность переключений между частотами для каждого соединения является псевдослучайной и известна только передатчику и приёмнику, которые каждые 312,5 мкс синхронно перестраиваются с одной несущей частоты на другую. Таким образом, если рядом работают несколько пар приёмник-передатчик, то они не мешают друг другу. Этот алгоритм является также составной частью системы защиты конфиденциальности передаваемой информации, поскольку переход происходит по псевдослучайному алгоритму и определяется отдельно для каждого соединения. При передаче цифровых данных и аудиосигнала (64 Кбит/с в обоих направлениях) используются различные схемы восстановления утраченных пакетов: аудиопакеты не повторяются (как правило), а цифровые данные – будут переданы повторно.

С 2002 года по настоящее время было разработано несколько версий Bluetooth, самая последняя версия Bluetooth 4.0 была утверждена в 2010 году и включает в себя три разновидности:

- классический Bluetooth со скоростью передачи 64 Кбит/с;
- высокоскоростной Bluetooth со скоростью передачи до 24 Мбит/с;
- Bluetooth с низким потреблением энергии и скоростью передачи в 1 Мбит/с.

Bluetooth с низким потреблением энергии предназначен для применения в миниатюрных электронных датчиках (температуры, давления, влажности, скорости передвижения и т.д.). Низкое энергопотребление достигается за счёт того, что передатчик включается только на время отправки пакета данных размером 8 – 27 байт, что обеспечивает возможность работы от одной батарейки в течение нескольких лет.

В остальных двух разновидностях Bluetooth используется усовершенствованный алгоритм коррекции ошибок и 128-разрядный ключ шифрования, что обеспечивает безопасность на относительно невысоком уровне.

Низкая безопасность соединения сопряжена с алгоритмом работы Bluetooth. Прежде чем начать обмен информацией, оба устройства должны сначала

найти друг друга, а затем установить соединение. Процесс поиска осуществляет активное устройство путём передачи пакетов определённого формата. Устройство, ответившее на такие пакеты, заявляет о себе, что оно находится в сети. Процесс установления соединения состоит из трёх фаз: генерации секретного ключа, генерации ключа связи и аутентификации соединения.

Для установления соединения одно устройство становится главным, а другое ведомым. Для вычисления секретного ключа главное устройство использует свой MAC-адрес и персональный идентификационный номер – Personal Identification Number (PIN), который известен обоим устройствам. По утверждённому стандартом Bluetooth алгоритму из этих двух величин главным устройством формируется некоторое 128-разрядное число и отсылается ведомому. Ведомое устройство получает, расшифровывает его и определяет, что активное устройство желает установить связь. В ответ на сообщение от активного устройства ведомое устройство генерирует по тому же закону 128-разрядное число со своим MAC-адресом и отсылает его активному устройству. Затем оба устройства обмениваются пакетами, содержащими 128-разрядные, случайным образом сгенерированные числа, которые в дальнейшем, будут выполнять роль ключей шифрования. После этого активное устройство посылает пакет с параметрами канала связи, а дождавшись ответа, посылает пакет синхронизации для настройки часов и коммутации частот. Далее каждый пакет передаётся на новой частоте, а алгоритм переключения частот определяется содержимым пакета синхронизации.

Низкая безопасность Bluetooth связана с тем, что существует возможность отправки злоумышленником особого сообщения в определённый момент, позволяющего начать процесс инициализации с устройством злоумышленника. Сама атака основана на уязвимостях в механизме аутентификации и создании ключа шифрации между двумя устройствами. В частности, в стандарте декларируется возможность потери ключа шифрации, поэтому устройство, утратившее ключ шифрации, посылает сообщение об его утрате. После чего начинается новый процесс установления соединения. В дальнейшем злоумышленник может на каждое зашифрованное сообщение, содержащее MAC-адрес и PIN-код, отправлять неправильный ответ, в результате после переполнения счётчика неудачных попыток начнётся новый процесс инициализации. Так будет продолжаться до тех пор, пока злоумышленник не подберёт PIN-код.

Проведённые в этом направлении исследования показывают, что время подбора PIN-кода зависит от его длины; так, при длине PIN-кода в 4 символа время подбора составляет всего 0,063 с, а при длине 5 символов – 0,75 с соответственно. Даже при длине PIN-кода в 6 символов время подбора составит

не более 7,6 с. Всё это справедливо при одном условии – злоумышленник находится рядом, на расстоянии не более 10 м.

Хотя передача информации происходит по принципу точка-точка, активное устройство может обслуживать до 7 ведомых устройств. Однако данная возможность практически не используется производителями. Кроме этого, из-за малой мощности передатчиков выпускаемых устройств, Bluetooth позволяет иметь устойчивую связь на расстоянии не более 10 м. В результате Bluetooth широко используется для связи сотовых телефонов между собой и сотовых телефонов с переносными компьютерами и там, где передаются небольшие объёмы информации.

14.1.2. Локальные беспроводные сети – Wi-Fi

Термин «Wi-Fi» изначально был придуман как игра слов, для привлечения внимания потребителя по аналогии с широко известным обозначением высококачественной аппаратуры воспроизведения звука Hi-Fi – High Fidelity (высокая точность (верность), в дальнейшем, с появлением аппаратуры воспроизведения звука ещё более высокого качества её стали обозначать High End – последняя высота).

Система беспроводной передачи цифровой информации, под маркой Wave LAN (волновая сеть передачи данных), была создана в 1991 году на базе стандарта компанией NCR, являющейся отделением компании AT&T, в Голландии и предназначалась для систем кассового обслуживания со скоростью передачи 1 или 2 Мбит/с. Компания NCR несколько раз переименовывалась, однако, несмотря на появление стандарта Radio Ethernet IEEE 802.11 в 1997 году и введение термина Wi-Fi, а затем стандарта IEEE 802.11a в 1999 году, так и не смогла конкурировать с проводными сетями и в 2004 году ушла с рынка беспроводных сетей.

Тем не менее, многие компании начали производство оборудования для Wi-Fi сетей. Однако стандарт IEEE 802.11a 1999 года хоть и регламентировал скорость передачи данных на уровне 54 Мбит/с, но использовал лицензируемый диапазон частот 5 ГГц, поэтому в этом же году был разработан новый стандарт IEEE 802.11b, регламентирующий две скорости передачи 5,5 и 11 Мбит/с, но использующий не лицензируемый диапазон частот 2,4 ГГц.

Продукты стандарта IEEE 802.11b, поставляемые разными изготовителями, тестируются на совместимость и сертифицируются организацией Wireless Ethernet Compatibility Alliance (WECA), которая в настоящее время больше известна под названием Wi-Fi Alliance. Совместимые беспроводные продукты, прошедшие испытания по программе "Альянса Wi-Fi", могут быть маркированы знаком Wi-Fi.

Долгое время стандарт IEEE 802.11b был очень распространённым, на базе него было построено большинство беспроводных сетей. Однако низкая скорость передачи информации при возросшей потребности потребителей привели к необходимости разработки новых стандартов. Так в 2002 году был утверждён стандарт IEEE 802.11g, предусматривающий использование диапазона частот 2,4 ГГц, обеспечивая скорость соединения 54 Мбит/с и гарантируя обратную совместимость со стандартом 802.11b.

В 2009 году был утверждён новый высокоскоростной стандарт IEEE 802.11n, теоретически обеспечивающий скорость передачи до 300 Мбит/с в диапазонах частот 2,4 и 5 ГГц и гарантирующий обратную совместимость со стандартами IEEE 802.11a/b/g. Оборудование, способное передавать информацию в двух диапазонах, может обеспечить скорость передачи до 600 Мбит/с. Также в версии стандарта n была частично решена проблема малой дальности связи и её качества – с помощью технологии MIMO (многоступенчатого приёма-передачи данных множеством антенн), которая требовала использования двух или трёх антенн. MIMO действует на основе математического вычисления разностного сигнала для группы антенн, что позволяет передать даже слабый сигнал с меньшими потерями и влиянием помех. Благодаря данным нововведениям стандарт позволяет передавать видео качества Full HD без потери пакетов, а также аудиопотоки.

Всё возрастающие объёмы передаваемой информации привели к необходимости создания нового, ещё более скоростного стандарта IEEE 802.11 – стандарта 802.11ac, утверждённого в 2012 году.

Отличий от стандарта IEEE 802.11n в новом стандарте не слишком много, но все они направлены на увеличение пропускной способности. Самое заметное – это расширение каналов MIMO с трёх до восьми. Восемь антенн теоретически позволяют удвоить пропускную способность до 800 Мбит/с. Кроме этого, в новом стандарте предусматривается увеличение ширины канала с 40 до 80 и 160 МГц, а это означает теоретически дополнительное увеличение пропускной способности в 2 и 4 раза соответственно (т.е. до 1,6 и 3,2 Гбит/с). Кроме того, была проведена оптимизация модуляции сигналов, что позволяет повысить скорость передачи до теоретических 5 Гбит/с.

В настоящее время уже появились устройства, обещающие пропускную способность до 1,3 Гбит/с на частоте 5 ГГц и до 450 Мбит/с на частоте 2,4 ГГц.

Однако обещания обещаниями, а достичь теоретической скорости передачи информации принципиально невозможно, и это связано не только с тем, что на радиосигнал влияют многочисленные факторы: электромагнитные поля, помехи от бытовой техники и электроники, препятствия на пути сигнала, отражения сигнала, даже солнечная активность, но и с особенностью работы

Wi-Fi, заключающейся в том, что весь спектр используемых частот разделяется на несколько каналов (узких полос частот), частично перекрывающих друг друга. Для нормальной же работы Wi-Fi необходимо, чтобы каналы не перекрывались.

В обоих частотных диапазонах 2,4 ГГц (полоса частот 2400 – 2483,5 МГц) и 5 ГГц (используются две полосы частот: полоса частот 5150 – 5240 МГц и полоса 5745 – 5825 МГц) ширина каждого частотного канала составляет 20 МГц, сдвинутых относительно друг друга на 5 МГц. В диапазоне 2,4 ГГц можно выделить четыре неперекрывающихся частотных канала с номерами 1, 6, 11 и 14 (рис. 14.2). Многие производители оборудования в настройках указывают данные частотные каналы по умолчанию. Можно использовать частотные каналы с частичным перекрытием, например каналы с номерами 1, 5, 9 и 13. В частотном диапазоне 5ГГц частично неперекрывающихся каналов – 23.

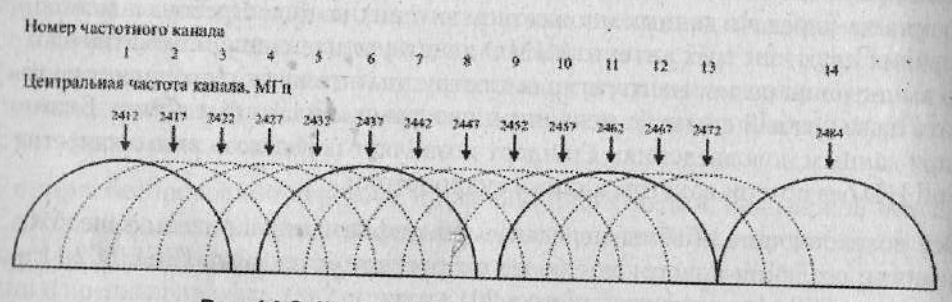


Рис. 14.2. Частотные каналы в диапазоне 2,4 ГГц

На скорость передачи влияет и используемый метод передачи в среде Ethernet – CSMA/CD. В беспроводном Ethernet применяется улучшенный метод доступа, позволяющий свести к минимуму конфликтные ситуации, основанный на передаче специальных сообщений для установления сеанса связи, и полудуплексный режим передачи. Как было отмечено выше, в каждом частотном диапазоне имеется несколько разрешённых частот для передачи. Узел может использовать любую частоту или строго фиксированную. В любом случае перед началом передачи информации узел, желающий передать сообщение, сначала прослушивает "эфир". Если на рабочей частоте не обнаружено активности, то узел сначала ожидает в течение некоторого случайного промежутка времени, потом снова "слушает эфир" и, если среда передачи данных все ещё свободна, осуществляет передачу. Наличие случайной задержки необходимо для того, чтобы исключить конфликты, при одновременном доступе нескольких узлов к сети на данном частоте. После этого узел начинает передачу информации. Если пакет приходит без искажений, то принимающий узел посылает обратно подтверждение. Целостность пакета

проверяется методом контрольной суммы. Получив подтверждение, передающий узел считает процесс передачи данного пакета завершённым. Если подтверждение не получено, то считается, что произошла коллизия, и пакет передается снова через случайный промежуток времени.

В условиях плохого качества связи узел может запросить очистку эфира, с целью прекращения передачи информации всеми узлами сети. В этом случае он посылает широковещательное сообщение – RTS (ready to send – готов к передаче), содержащее информацию о продолжительности предстоящей передачи и адресате. Это сообщение позволяет другим узлам задержать передачу на время объявленной длительности передачи, тем самым освободить эфир. Если среда свободна и приёмник готов принять сообщение, то он отвечает передающему узлу сообщением CTS (clear to send – свободен для передачи).

Для передачи потоковых данных применяется централизованное управление, при этом точка доступа последовательно опрашивает узлы на предмет желания передачи потоковых данных. Если какой-нибудь из узлов сообщает, что он будет передавать потоковые данные, то точка доступа выделяет ему промежуток времени, в который все остальные узлы должны прекратить передачу. Предоставление монопольного режима передачи информации снижает эффективность работы беспроводной сети.

Поскольку узлы сети могут быть мобильными устройствами с автономным питанием, то в стандарте IEEE 802.11 большое внимание уделено вопросам управления питанием. В частности, предусмотрен режим, когда узел через определённые промежутки времени "просыпается", чтобы принять сигнал включения, который, возможно, передает точка доступа. Если этот сигнал принят, клиентское устройство включается, в противном случае оно снова "засыпает" до следующего цикла приема информации.

В Wi-Fi может применяться несколько типов соединений, начиная от «точка-точка» до «маршрутизатор-компьютеры». Наиболее широко на практике применяется метод доступа отдельных компьютеров через точку доступа к проводной сети. Точка доступа может быть интегрирована в маршрутизатор. В любом случае доступ к точке доступа может быть организован путём статической или динамической настройки радиоканалов.

Статичная настройка предполагает принудительное присвоение определённого радиоканала нескольким узлам и связь с определённой точкой. Такой метод целесообразно применять при наличии большого количества узлов, разделив их на независимые группы. В этом случае настройку необходимо производить как самой точки доступа, так и узлов.

Динамическая настройка предполагает возможность передачи информации узлами по любому свободному, в данный момент времени, радиоканалу. Установка нескольких точек доступа с перекрытием приводит к тому, что узел имеет возможность доступа сразу к нескольким точкам. Выбор точки доступа для работы осуществляется автоматически на основе уровня принимаемого сигнала, чем он выше, тем больше вероятность, что будет выбрана данная точка. Данный метод чаще всего используется при организации открытого доступа к Internet, например, в кафе, гостиницах и т.д.

Для улучшения качества связи в Wi-Fi предусмотрено принудительное понижение скорости передачи. При ухудшении качества связи, о чём свидетельствует некоторое количество неправильно принятых пакетов подряд, передатчик последовательно, с шагом в 1 МГц, понижает скорость передачи до достижения приемлемого количества неправильно принятых пакетов. При достижении скорости в 1 Мбит дальнейшее понижение осуществляется с шагом в 100 КГц. Минимальная скорость передачи составляет 100 Кбит/с.

Как и любая компьютерная сеть, Wi-Fi является источником повышенного риска от несанкционированного доступа, поскольку проникнуть в беспроводную сеть значительно проще, чем в обычную, – не нужно подключаться к проводам, достаточно только оказаться в зоне приёма сигнала.

Для защиты беспроводных сетей разработано достаточное количество методов и способов. Наибольшую защиту обеспечивает применение сразу нескольких методов, на трёх основных наиболее опасных направлениях: аутентификация пользователя, целостность и конфиденциальность передаваемой информации. В частности, конфиденциальность информации обеспечивается её шифрованием с использованием 128-разрядных ключей, сгенерированных случайным образом и динамически меняющихся через каждые 0000 переданных пакетов.

Дальность связи, как и скорость передачи, в Wi-Fi определяется множеством факторов. При декларируемой дальности передачи информации от 120 до 00 м на открытой местности реальная дальность в помещениях не превышает 30 – 50 м.

14.1.3. Беспроводные сети города – WiMAX

Аббревиатура WiMAX происходит от первых букв выражения World-wide Interoperability for Microwave Access (всемирное взаимодействие в микроволновом доступе) и впервые появилась в 2001 году. WiMAX представляет собой телекоммуникационную технологию для предоставления беспроводной связи на большие расстояния, разрабатываемую в рамках стандарта IEEE 802.16. Разработчики стандарта искали оптимальные решения как

для фиксированного, так и для мобильного применения, но совместить все требования в рамках одного стандарта им не удалось, что привело к созданию двух отдельных версий стандарта (вернее, их можно считать двумя разными стандартами). Каждая из версий WiMAX определяет свои рабочие диапазоны частот, ширину полосы пропускания, мощность излучения, методы передачи и доступа, способы кодирования и модуляции сигнала, принципы повторного использования радиочастот и другие показатели, потому WiMAX-системы, основанные на версиях стандарта IEEE 802.16 e и d, практически несовместимы.

Основное различие двух технологий состоит в том, что фиксированный WiMAX позволяет обслуживать только статических абонентов, а мобильный ориентирован на работу с пользователями, передвигающимися со скоростью до 150 км/ч. Мобильность означает наличие функций роуминга и «бесшовного» переключения между базовыми станциями при передвижении абонента.

Фиксированный WiMAX определяется стандартом IEEE 802.16d, утверждённым в 2004 году. Устройства этого стандарта используют ортогональное частотное уплотнение в пределах прямой видимости или её отсутствии. Пользовательские устройства представляют собой стационарные модемы для установки вне и внутри помещений, а также PCMCIA-карты для ноутбуков. Под него отведены диапазоны 3,5 и 5 ГГц. Стандарт определяет скорость передачи до 75 Мбит/с и дальность передачи от 25 до 80 км.

Мобильный WiMAX определяется стандартом IEEE 802.16e, утверждённым в 2005 году, и оптимизирован для поддержки мобильных пользователей. Применяется масштабируемый OFDM-доступ (SOFDMA), возможна работа при наличии либо отсутствии прямой видимости. Планируемые частотные диапазоны для сетей мобильного WiMAX: 2,3 – 2,5; 2,5 – 2,7 и 3,4 – 3,8 ГГц. Стандарт определяет скорость передачи до 40 Мбит/с и дальность передачи от 1 до 5 км.

В стандарте WiMAX используется метод передачи данных по принципу «точка-точка» по технологии множественного доступа с разделением по времени, согласно которой базовая станция выделяет абонентским станциям временные интервалы, чтобы они могли передавать данные в определённой очередности, а не случайным образом. Для реализации дуплексного режима обмена данными используются две технологии: дуплексный режим с разделением по времени и дуплексный режим с разделением по частотам. Для предотвращения несанкционированного доступа и защиты пользовательских данных осуществляется шифрование всего передаваемого по сети трафика.

В 2012 году утверждён новый стандарт WiMAX-2 – IEEE 802.16m, определяющий скорость передачи до 1 Гбит/с для стационарных устройств и до 100 Мбит/с для мобильных устройств, при дальности передачи от 120 до 150 км.

Несмотря на явные преимущества технологии WiMAX, основным тормозом в её широком внедрении является высокая стоимость оконечного оборудования пользователя и невысокая дальность передачи информации. А также, несмотря на декларируемые стандартом расстояния до 80 км, производимое оборудование пока не может перешагнуть рубеж в 5 – 10 км, то же самое относится и к скорости передачи. Поэтому основными потребителями оборудования WiMAX являются корпоративные клиенты, которым необходим более дешёвый канал связи, нежели проводной.

14.2. Сети мобильной связи

Мобильная односторонняя телефонная радиосвязь впервые появилась в США в 1921 году, а в 1933 была внедрена двухсторонняя телефонная связь, во всех этих системах использовалась амплитудная модуляция. Начиная с 1940 года стали применять частотную модуляцию.

В конце 50-х годов прошлого века в СССР начинается разработка системы автомобильного радиотелефона «Алтай». В 1957 году московский инженер Л.И. Куприянович создал опытный образец носимого автоматического дуплексного мобильного радиотелефона ЛК-1 и базовую станцию к нему. Мобильный радиотелефон весил около трех килограммов и имел радиус действия 20 – 30 км. В 1958 году он создаёт усовершенствованные модели аппарата весом 0,5 кг и размером с папиросную коробку.

Первая в мире автоматическая система мобильной связи «Алтай» была запущена в эксплуатацию в 1963 в Москве. Мобильный телефон «Алтай» представлял собой полноценный телефон, устанавливаемый в автомобиле. Аналогичная система в США была запущена в коммерческую эксплуатацию только 6 лет спустя – в 1969 году. Между тем в СССР к 1970 году система «Алтай» была установлена и успешно работала уже в 30 городах в диапазоне 330 МГц.

Таким образом, к 1970 году подвижная телефонная радиосвязь, с одной стороны, уже получила достаточно широкое распространение, но с другой – явно не успевала за быстро растущими потребностями, при ограниченном числе каналов в жёстко определённых полосах частот. Выход был найден в виде системы сотовой связи, что позволило резко увеличить ёмкость за счёт повторного использования частот в системе с ячеистой структурой.

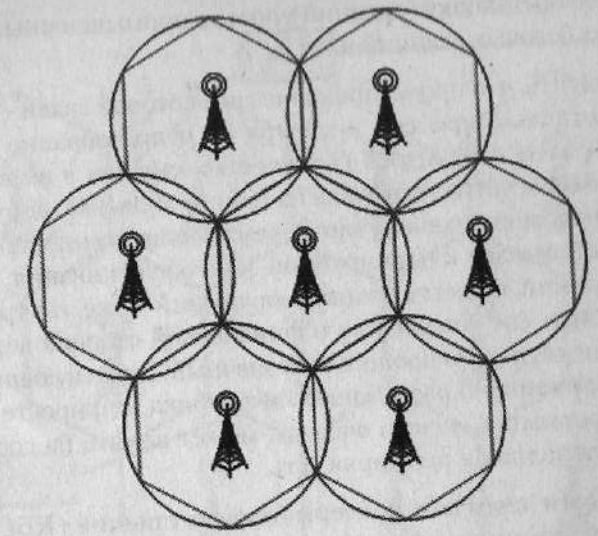


Рис. 14.3. Соты в системе сотовой связи

Для построения мобильной связи необходимо обеспечить достаточную площадь покрытия, чтобы абоненты сети могли беспрепятственно перемещаться в пространстве. При этом необходимо установить станции приёма-передачи (с ограниченной дальностью передачи) таким образом, чтобы зоны действия станций перекрывались для исключения пропусков – «мёртвых зон», в которых связь невозможна.

Передающая станция в пространстве распространяет радиоволны вокруг себя по окружности, при условии, что на их пути нет никаких препятствий. Поэтому самым оптимальным расположением радиостанций в пространстве, с точки зрения площади покрытия и минимального пересечения окружностей, является шестиугольник, как показано на рис. 14.3. Для покрытия большой площади необходимо устанавливать достаточно большое количество антенн, с пересечением по линиям вписанного шестиугольника. Такое расположение антенн и зоны их перекрытия напоминают пчелиные соты, поэтому этот тип связи назвали сотовой.

В центре окружности – рабочей зоне (соте) – устанавливают так называемую базовую станцию (БС), которая осуществляет связь по радиоканалам со многими абонентскими станциями (пользователями), установленными на подвижных объектах (автомобилях, у Вас в кармане) и находящихся в е рабочей зоне. Для каждой соты в рамках всей сети при проектировании про водится комплекс специальных мероприятий, в том числе и частотное пла нирование, которое учитывает условия распространения радиоволн

общую частотную обстановку (средний уровень промышленных помех, сигналы от смежных базовых станций и т.п.).

Необходимо заметить, что проектирование сети сотовой связи – весьма трудоемкий и кропотливый процесс, несмотря на использование компьютерных систем. Учёт всех параметров (количество каналов в выделенном частотном диапазоне, расчётная нагрузка на одного абонента, допустимая интенсивность потерь, минимальная приемлемая напряженность поля) и их топографическая привязка к планируемой зоне обслуживания, опять же, с учётом типа местности, существующих сооружений – всё это требует больших трудовых затрат (не говоря уже о финансовой стороне дела). К тому же проектирование сети – это процесс бесконечный. Действующие сегменты сети выдают информацию о распределении трафика и приросте числа абонентов, и эта информация, в свою очередь, может влиять на составленные раньше проекты, дополняя и расширяя сеть.

Базовые станции сети через контроллеры базовых станций (КБС) соединены проводными телефонными или радиорелейными линиями связи с центральной станцией данного региона (города, области) – (центр коммутации подвижной связи), которая обеспечивает соединение мобильных абонентов с абонентами телефонной сети общего пользования с помощью коммутационных устройств (рис. 14.4). Таким образом, центр коммутации (ЦК) обслуживает группу сот, осуществляя при этом переключение каналов в соте, эстафетную передачу абонентов между сотами, формирует данные, необходимые для выписки счетов за предоставленные сетью услуги связи, накапливает данные по состоявшимся разговорам и передает их в центр расчетов – биллинг-центр (рис. 14.5).

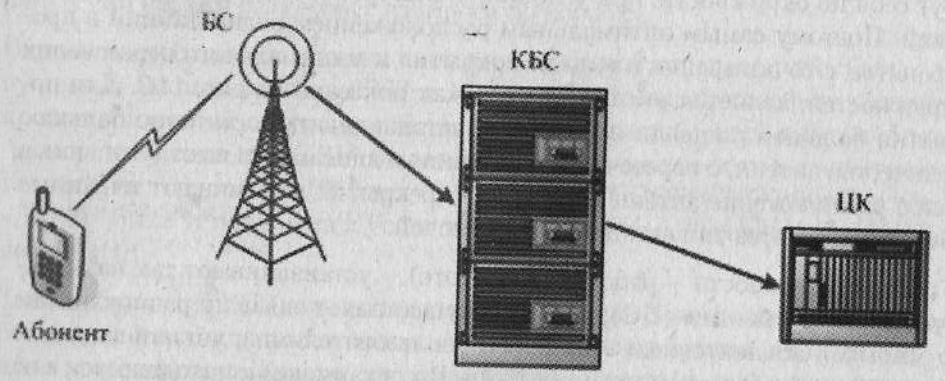


Рис. 14.4. Принцип передачи в системе сотовой связи

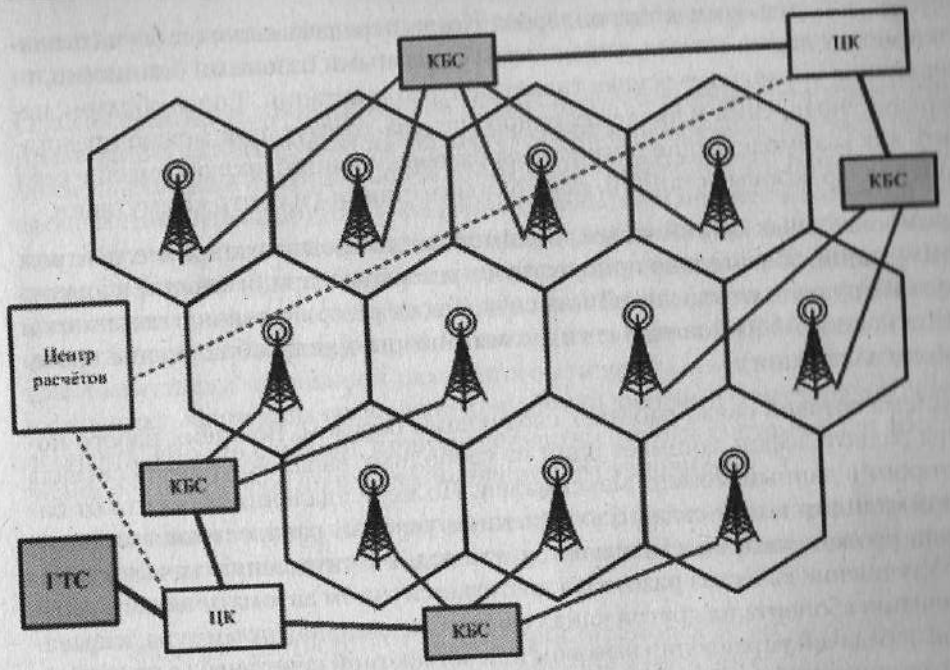


Рис. 14.5. Структура сотовой связи

Центр коммутации составляет статистические данные, необходимые для контроля работы и оптимизации сети, и участвует не только в управлении вызовами, но также управляет процедурами регистрации местоположения и передачи управления. Регистрация местоположения подвижных станций (абонентов) необходима для обеспечения доставки вызова перемещающимся подвижным абонентам от абонентов телефонной сети общего пользования или других подвижных абонентов. Процедура передачи вызова позволяет сохранять соединения и обеспечивать ведение разговора, когда подвижная станция (абонент) перемещается из одной зоны обслуживания в другую.

Таким образом, основным элементом сотовой сети любого стандарта является базовая станция, которая занимается приёмом звонков абонентов и передачей данных по радиоканалу. В зависимости от стандарта связи, базовые станции работают в диапазоне частот от 450 до 2100 МГц, они составляют основу ячеек – сот. Поскольку радиус работы таких станций составляет порядка 10-12 км за городом и около 3-5 км в городе, их строят много и располагают относительно недалеко друг от друга. Полностью автономные и автоматизированные базовые станции представляют собой небольшие контейнеры, которые устанавливаются, как правило, на крыше зданий. Передача вызовов в сотах, управляемых одним контроллером базовых станций,

осуществляется этим контроллером. Когда передача вызовов осуществляется между двумя сетями, управляемыми разными базовыми станциями, то первичное управление осуществляет центр коммутации. Таким образом, иерархию управления в пределах одного центра коммутации можно представить как взаимодействие подсистемы базовых станций, включающей в себя контроллер базовых станций, сами базовые станции и центр коммутации.

Кроме основных линий связи, соединяющих базовые станции с центром коммутации, обязательно присутствуют резервные, принимающие абонентскую нагрузку в случае аварии на себя. Также резервированию подлежит и всё основное оборудование сети (коммутаторы, каналообразующие системы, сигнализация).

Система сотовой связи работает следующим образом. Во время набора номера радиотелефон занимает один из свободных каналов, уровень сигнала которого в данный момент максимален. По мере удаления абонента от базовой станции или в связи с ухудшением условий распространения радиоволн уровень сигнала уменьшается, что ведёт к ухудшению качества связи. Улучшение качества разговора достигается путём автоматического переключения абонента на другой канал связи. Специальная процедура, называемая передачей управления вызовом или эстафетной передачей (в иностранной технической литературе – handover или handoff), позволяет переключить разговор на свободный канал другой базовой станции, в зоне действия которой оказался в это время абонент.

Аналогичные действия предпринимаются при снижении качества связи из-за влияния помех или при возникновении неисправностей коммутационного оборудования. Для контроля таких ситуаций базовая станция снабжена специальным приёмником, периодически измеряющим уровень сигнала сотового телефона разговаривающего абонента и сравнивающим его с допустимым пределом. Если уровень сигнала меньше этого предела, то информация об этом автоматически передается в центр коммутации по служебному каналу связи. Центр коммутации выдаёт команду об измерении уровня сигнала сотового радиотелефона абонента на ближайшие к нему базовые станции. После получения информации от базовых станций об уровне этого сигнала центр коммутации переключает радиотелефон на ту из них, где уровень сигнала оказался наибольшим. Это происходит за время от 250 до 1250 мс, поэтому абонент почти не замечает переключения.

Необходимо заметить, что в цифровых системах сотовой связи время переключения меньше, чем в аналоговых. А в цифровом стандарте CDMA оно вообще отсутствует как таковое в силу использования технологии "soft handover", при которой до момента переключения и некоторое время после него мобильной станции предоставляется одновременно два канала (старый и

новый) связи, организуя тем самым своеобразный неразрывный во времени канальный "шлюз".

Одна из важных услуг сети сотовой связи - предоставление возможности использования одного и того же радиотелефона при поездке в другой город, область или даже страну, причём сотовая сеть позволяет не только самому абоненту звонить из другого города или страны, но и получать звонки от тех, кто не успел застать его дома. В сотовой радиосвязи такая возможность называется роуминг (от английского слова roam – скитаться, блуждать). Для организации роуминга сотовые сети должны быть одного стандарта (телефон стандарта GSM не будет работать в сети стандарта CDMA и т. п.), а центры коммутации подвижной связи этого стандарта должны быть соединены специальными каналами связи для обмена данными о местонахождении абонента. Иными словами, применительно к сотовым системам для обеспечения роуминга необходимо выполнение трёх условий:

- наличие в требуемых регионах сотовых систем стандарта, совместимого со стандартом компании, у которой был приобретён радиотелефон;
- наличие соответствующих организационных и экономических соглашений об обслуживании абонентов в роуминге;
- наличие каналов связи между системами, обеспечивающими передачу звуковой и другой информации для абонентов в роуминге.

При перемещении абонента в другую сеть её центр коммутации запрашивает информацию в первоначальной сети и при наличии подтверждения полномочий абонента регистрирует его. Данные о местоположении абонента постоянно обновляются в центре коммутации первоначальной сети, и все поступающие туда вызовы автоматически переадресовываются в ту сеть, где в данный момент находится абонент.

При организации роуминга недостаточно провести только технические мероприятия по соединению различных сетей сотовой связи. Очень важно еще решить проблему взаиморасчётов между операторами этих сетей.

Различают три вида роуминга:

1. *Автоматический* (именно с этой формой за рубежом обычно и связывают понятие роуминга), то есть предоставление абоненту возможности выйти на связь в любое время в любом месте.
2. *Полуавтоматический*, когда абоненту для пользования данной услугой в каком-либо регионе необходимо предварительно поставить о этом в известность своего оператора.

3. *Ручной*, по сути, простой обмен одного радиотелефона на другой, подключенный к сотовой системе другого оператора.

Наиболее важным и сложным, с точки зрения реализации в любой системе сотовой связи, является интерфейс между базовой и мобильной станциями. Наибольшую сложность вызывает необходимость в ограниченном частотном диапазоне, при воздействии помех, передать как можно больше информации от как можно большего числа абонентов. Один стандарт сотовой связи от другого отличаются в первую очередь именно принципами организации радиоинтерфейса. От того, насколько эффективно будет решена вышеуказанная проблема, и зависит успех той или иной системы сотовой связи.

Весь частотный диапазон в окружающем пространстве является общедоступным ресурсом, к которому теоретически каждый может получить доступ. Однако при одновременной передаче в одном и том же частотном диапазоне произойдет то же самое, как если бы по одному телефону пытались разговаривать одновременно несколько абонентов – произойдёт наложение сигналов, или интерференция. Интерференция обуславливает ухудшение качества соединения, поскольку постоянно будут возникать ошибки, что, в конечном счете, может привести к обрыву соединения. Чтобы не допустить возникновения интерференции, весь частотный диапазон разделяют на диапазоны (полосы). В сотовой связи используются диапазоны 800 МГц, 900 МГц, 1800 МГц, 1900 МГц и 2100 МГц и некоторые другие, в зависимости от стандарта и страны. Цифра в названии диапазона представляет собой условное название и указывает на частоту, в районе которой располагаются используемые частоты.

Не все частотные диапазоны одинаковы. Так, частоты диапазона 900 МГц меньше ослабляются при прохождении стен, крыш домов и других преград. Чтобы достичь таких же характеристик в диапазоне 1800 МГЦ, необходимо устанавливать усилители большей мощности на базовых станциях.

Практически все сервисы, предоставляемые сотовыми системами связи, двусторонние, они предполагают не только передачу, но и приём информации. Для их нормальной и устойчивой работы необходима организация двух каналов передачи от базовой станции к мобильной и наоборот, разнесённых на 45 – 80 МГц. Канал связи от базовой станции к мобильной получил название downlink (DL), а от мобильной станции к базовой – uplink (UL). Обычно во всех системах связи эти каналы одинаковы по максимальной скорости передачи данных и ширине частотного канала.

После получения частотного диапазона и разделения его на два канала UL и DL следующей важной задачей является разграничение доступа между абонентами в сеть. Для этого существует несколько технологий множественного доступа: frequency division multiple access – FDMA (множественный доступ с частотным разделением каналов), time division multiple access – TDMA (множественный доступ с временным разделением каналов), code division multiple access – CDMA (множественный доступ с кодовым разделением каналов), а также их разновидности и сочетания.

Итак, все частотные диапазоны делятся на два канала – канал доступа к базовой станции и канал доступа к абоненту. Как было отмечено выше, эти каналы разнесены на 45 – 80 МГц для исключения интерференции и содержат несколько разрешённых частот для передачи информации.

- В диапазоне частот 450 МГц ширина канала составляет 30 КГц, а выделенный диапазон частот лежит в пределах 453 – 457,5 МГЦ для канала UL и 463 – 467,5 МГц для канала DL, что позволяет организовать до 150 каналов аналоговой связи.
- В диапазоне частот 800 МГц ширина канала составляет 30 КГц, а выделенный диапазон частот лежит в пределах 824 – 849 МГЦ для канала UL и 869 – 894 МГц для канала DL, что позволяет организовать 533 канала.
- В диапазоне частот 900 МГц ширина канала составляет 200 КГц, а выделенный диапазон частот лежит в пределах 890 – 915 МГЦ для канала UL и 935 – 960 МГц для канала DL, что позволяет организовать 124 канала.
- В диапазоне частот 1800 МГц ширина канала составляет 200 КГц, а выделенный диапазон частот лежит в пределах 1710 – 1785 МГЦ для канала UL и 1805 – 1880 МГц для канала DL, что позволяет организовать 374 канала.
- В диапазоне частот 1900 МГц ширина канала составляет 200 КГц, а выделенный диапазон частот лежит в пределах 1850 – 1910 МГЦ для канала UL и 1930 – 1990 МГц для канала DL, что позволяет организовать 300 каналов.

Естественно, что такого количества каналов явно недостаточно для обеспечения связи между большим количеством абонентов. Кроме этого, каждому оператору связи выделяется определённое количество каналов в каждом диапазоне. Для увеличения количества абонентов кроме разделения частот применяют метод временного разделения каждого канала. В этом случае в рамках каждой частоты выделяется 8 циклически повторяющихся временных интервалов (в терминах сотовой связи – слот). В результате для каждого абонента в соте предоставляется свой временной интервал, в рамках которого он может передавать информацию в каком-то одном частотном канале

Это позволяет увеличить количество абонентов в 8 раз.

Ещё большего увеличения количества абонентов можно добиться, используя метод кодового разделения каналов. При этом каждому абоненту в соте присваивается свой уникальный, в данной соте, код, далее источник предваряет пакет передаваемой информации этим кодом и передаёт его в эфир, на одной из действующих в данной конкретной соте частоте. Приёмник знает код источника и может извлечь из общего потока информацию, предназначенную именно для него. Для передачи информации отводится интервал в 15/26 мс, что примерно соответствует 0,577 мс. За одну секунду около 1700 станций могут передать свои сообщения.

В стандарте GSM (Global System for Mobile Communications), например, используется сочетание методов FDMA и TDMA, что позволяет значительно увеличить количество абонентов сети.

В настоящее время принято говорить о поколениях (generation) сотовой связи, которые принято обозначать в виде 1G ... 4G, где цифра обозначат поколение. К сожалению, чёткие границы между поколениями стали стираться (исключение составляет переход от поколения 1G к 2G, поскольку связан с переходом от аналоговой связи к цифровой) в связи с тем, что поколения друг от друга явно отличаются, в основном, количеством предоставляемых услуг и неявно декларируемой скоростью передачи информации.

Для пользователя все поколения выше 2G цифровые и принципиально друг от друга не отличаются, а добавление новых типов сервисов им воспринимается как само собой разумеющееся развитие технологии связи, поэтому деление сотовой связи на поколения является чисто маркетинговым ходом для обоснования повышенной стоимости оказываемых услуг и аппаратных устройств, поддерживающих данные услуги соответственно. В результате поставщиками оборудования были выделены поколения не только 1G, 2G и 3G, но промежуточные – типа 2,5G, 2,75G и 3,5G.

Новое поколение сотовой связи – 4G – должно обеспечивать декларируемую скорость передачи информации в 1 Гбит/с для стационарного оборудования и в 100 Мбит/с для мобильного. Естественно, что для пользователя совершенно неважно, какие технологические новшества для этого потребуются (более сложные методы модуляции, пакетные передачи и т.д.). Как показывает практическое применение беспроводной связи, декларируемая скорость передачи, особенно у быстроходных стандартов, обычно в 10 – 30, а то и в 50 раз ниже реальной. Поэтому не стоит слишком уповать на новый стандарт, однако любое повышение реальной скорости передачи всегда приветствуется всеми пользователями.

И последнее замечание относительно беспроводной связи и сотовой в частности. С целью сокращения расходов на передающее оборудование целесообразно всем провайдерам договориться об использовании существующего оборудования друг друга. А при необходимости строительства новых сот организовать отдельную компанию, решающую весь комплекс проблем, связанных как со строительством, так и с обслуживанием приёмо-передающего оборудования. В настоящее же время каждый провайдер самостоятельно устанавливает передающие базовые станции, контроллеры базовых станций и центры коммутации, поэтому частенько на одном здании можно видеть целый лес сотовых антенн.

Контрольные вопросы

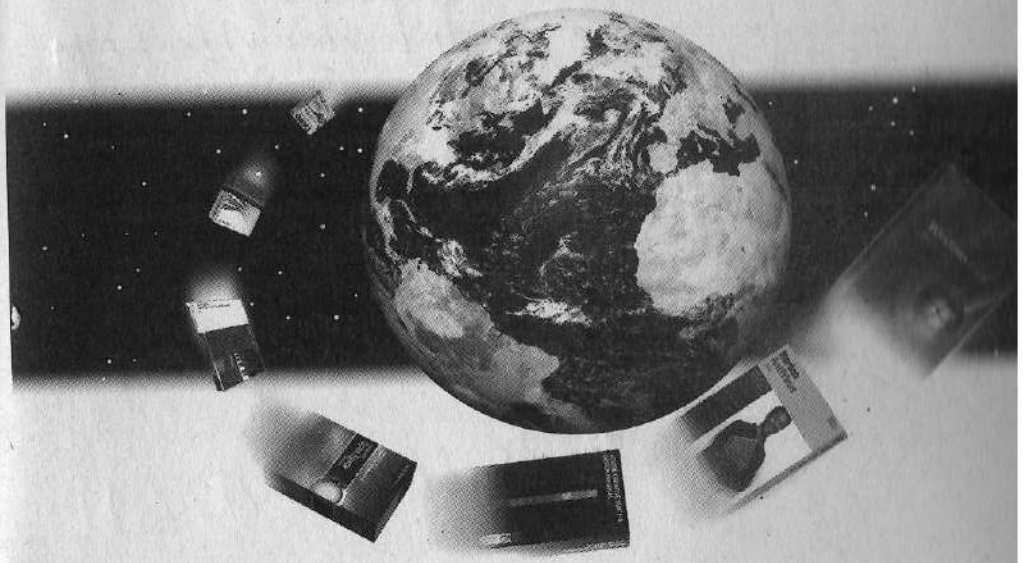
1. *Назовите причины, которые способствовали распространению беспроводной связи.*
2. *Что такое радиорелейная связь?*
3. *Какие принципы положены в основу тропосферной связи?*
4. *Что представляет собой станция тропосферной связи?*
5. *Назовите основные преимущества персональных беспроводных сетей Bluetooth.*
6. *Объясните принцип работы системы Bluetooth.*
7. *Поясните, как осуществляется процесс передачи информации между узлами в системе Bluetooth.*
8. *Назовите основные недостатки персональных беспроводных сетей Bluetooth.*
9. *Что означает аббревиатура Wi-Fi?*
10. *Какие диапазоны частот использует Wi-Fi?*
11. *Какое количество узлов можно подключить к одной точке Wi-Fi?*
12. *Какой метод доступа к передающей среде используется в Wi-Fi?*
13. *Какова минимальная скорость передачи информации в Wi-Fi?*
14. *Назовите основные преимущества сетей Wi-Fi.*
15. *Назовите основные недостатки сетей Wi-Fi.*
16. *Почему возникла необходимость в появлении WiMAX?*
17. *Каковы декларируемые характеристики WiMAX?*
18. *Назовите основные преимущества сетей WiMAX.*
19. *Назовите основные недостатки сетей WiMAX.*

20. В каком году и где появилась первая в мире автоматическая система мобильного телефона?
21. Почему мобильную связь называют сотовой?
22. Какие основные составляющие содержит система сотовой связи?
23. Какие частотные диапазоны отведены для сотовой связи?
24. Каким образом происходит переход от одной соты к другой?
25. Зачем нужен роуминг и как он реализуется?
26. Назовите методы разграничения доступа между абонентами при входе в сеть.
27. Какое количество каналов содержит каждый диапазон?
28. Какие существуют методы увеличения количества мобильных пользователей при ограниченном количестве каналов?
29. Какие поколения сотовой связи вам известны?
30. Какие характеристики декларирует стандарт 4G?

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ: ЛИЦЕНЗИРОВАНИЕ • ОБУЧЕНИЕ • КОНСАЛТИНГ

softline®

Софт со всего света



Сотрудничайте с нами в 50 городах 15 стран:

Москва
Санкт-Петербург
Архангельск
Барнаул
Владивосток
Волгоград
Воронеж
Екатеринбург
Ижевск
Иркутск
Казань
Калининград
Кемерово
Краснодар
Красноярск
Набережные Челны
Нижний Новгород
Новосибирск
Омск
Оренбург
Пермь
Ростов-на-Дону
Самара
Саратов
Сыктывкар
Томск
Тюмень
Ульяновск
Уфа
Хабаровск
Челябинск
Ярославль
Минск
Гомель
Витебск
Киев
Харьков
Алматы
Астана
Ашгабад
Бишкек
Баку
Душанбе
Ереван
Тбилиси
Ташкент
Караганда
Стамбул
Тегеран
Улан-Батор

www.softline.ru

Санкт-Петербург,
ул. Большая Монетная, д.16,
корп.5, лит.Е, оф.202
E-mail: info.spb@softline.ru

(812) 336-44-46