ГЛАВА 10

Основы маршрутизации и принципы построения подсетей

В этой главе...

- описано назначение маршрутизируемых протоколов и протоколов маршрутизации, как, например, механизмы протокола IP;
- рассмотрены функции протокола IP и проведено сравнение механизмов передачи данных с установлением и без установления соединения;
- объясняется работа маршрутизаторов на третьем уровне модели OSI;
- описан формат сообщений и назначение полей в сообщениях протоколов маршрутизации;
- проведен сравнительный анализ коммутации второго уровня и маршрутизации третьего в модели OSI;
- дано определение протокола маршрутизации, описаны механизмы поиска наилучшего маршрута для потока данных и рассмотрены механизмы, которые обеспечивают актуальность таблиц маршрутизации;
- описаны различия между статической и динамической маршрутизацией;
- объясняется, как маршрутизаторы используют функции выбора маршрута и коммутации для передачи пакетов через объединенные сети;
- описаны различия дистанционно-векторных протоколов маршрутизации и протоколов маршрутизации с учетом состояния каналов, а также рассмотрены особенности их конвергенции;
- рассмотрены различия протоколов маршрутизации внешних и внутренних шлюзов, а также приведены примеры каждого из типов протоколов;
- объяснены особенности и преимущества разбиения сетей на подсети;
- описано, как создавать подсети с помощью сетевой маски и рассчитывать количество узлов и сетей;
- Показано, как рассчитать адрес сети с использованием логической операции AND.

Ключевые определения главы

Ниже перечислены основные определения главы, полные расшифровки терминов приведены в конце:

протокол, с. 491,
маршрутизируемый протокол, с. 491,
пакет, с. 491,
протокол маршрутизации, с. 492,
IP-адрес, с. 493,
протокол без установления соединения, с. 494,
дейтаграмма, с. 494,
домен коллизий, с. 496,
система доставки с установлением соединения, с. 499,
маршрутизатор, с. 501,
метрика маршрутизации, с. 502,
широковещательный домен, с. 504,
подсети, с. 504,
таблица маршрутизации, с. 505,
MAC-адрес, с. 513,
широковещательные рассылки, с. 514,
счетчик транзитных узлов, с. 516,
алгоритмы маршрутизации, с. 517,
протоколы внутренних шлюзов, с. 519,
протоколы внешних шлюзов, с. 519,
автономная система, с. 519,
алгоритм дистанционно-векторной маршрутизации, с. 520,
протокол маршрутной информации, с. 521,
протокол маршрутизации внутреннего шлюза, с. 521,
усовершенствованный протокол маршрутизации внутреннего шлюза, с. 521,
алгоритм с учетом состояния каналов, с. 521,
бесклассовая междоменная маршрутизация, с. 525,
октет, с. 527,
адрес подсети, с. 528.

Эта глава посвящена вопросам, связанным с работой фундаментального протокола сети Internet (Internet Protocol — IP). В ней обсуждаются вопросы доставки сообщений IP, процессы модификации заголовка устройствами третьего уровня и фактическая структура IP-пакета. В главе также обсуждается связь между сетевыми службами с установлением и без установления соединения и объясняется разница между протоколами маршрутизации и маршрутизируемыми протоколами, а также механизмы, используемые маршрутизаторами для определения расстояния между удаленными точками. Глава ознакомит читателя с технологиями маршрутизации по вектору расстояния (distance-vector), состоянию канала (link-state) и гибридной (hybrid) технологией и расскажет о том, как каждая из них решает общие проблемы маршрутизации.

Обратите внимание на относящиеся к главе интерактивные материалы, которые находятся на компакт-диске, предоставленном вместе с книгой: электронные лабораторные работы (e-Lab), видеоролики, мультимедийные интерактивные презентации (PhotoZoom). Эти вспомогательные материалы помогут закрепить основные понятия и термины, изложенные в этой главе.

Маршрутизируемые протоколы

Протокол IP (Internet-протокол) является маршрутизируемым протоколом сети Internet. Пакеты маршрутизируются по оптимальному пути от отправителя к получателю на основе уникальных идентификаторов — IP-адресов. Данные могут быть правильно доставлены получателю в том случае, если в сети требуемым образом работают механизмы пересылки пакетов, устройства, преобразующие данные из одного формата в другой, а также протоколы с установлением и без установления соединения. В первой части главы основное внимание уделено именно перечисленным выше компонентам сети передачи данных.

Маршрутизируемые и маршрутизирующиеся протоколы

Протоколом называется основанный на стандартах набор правил, определяющий принципы взаимодействия компьютеров в сети. Протокол также задает общие правила взаимодействия разнообразных приложений, сетевых узлов или систем, создавая таким образом единую среду передачи. Взаимодействующие друг с другом компьютеры обмениваются данными; чтобы принять и обработать сообщения с данными, компьютерам необходимо знать, как сформированы сообщения и что они означают. Примерами использования различных форматов сообщений в разных протоколах могут служить установление соединения с удаленной машиной, отправка сообщений по электронной почте или передача файлов и данных; интуитивно понятно, что разные службы используют разные сообщения.

Протокол описывает:

- формат сообщения, которому приложения обязаны следовать;
- способ обмена сообщениями между компьютерами в контексте определенного действия, такого, как отправка сообщений по сети.

Схожее звучание терминов “маршрутизируемый протокол” и “протокол маршрутизации” нередко приводит к путанице. Приведенные ниже определения помогут прояснить ситуацию.

- *Маршрутизируемый протокол* — это любой сетевой протокол, адрес сетевого уровня которого предоставляет достаточное количество информации для доставки пакета от одного сетевого узла другому на основе используемой схемы адресации. Такой протокол задает форматы полей внутри *пакета*. Пакеты обычно передаются от одной конечной системы другой. Маршрутизируемый протокол использует таблицу маршрутизации для пересылки пакетов. Примеры маршрутизируемых протоколов приведены на рис. 10.1. В их число входят:

- Internet-протокол (IP);
- протокол межсетевого пакетного обмена (Internetwork Packet Exchange — IPX);
- протокол AppleTalk.

Легче всего запомнить, что такое маршрутизируемые протоколы, если помнить, что это протоколы, которые связаны с передачей данных.

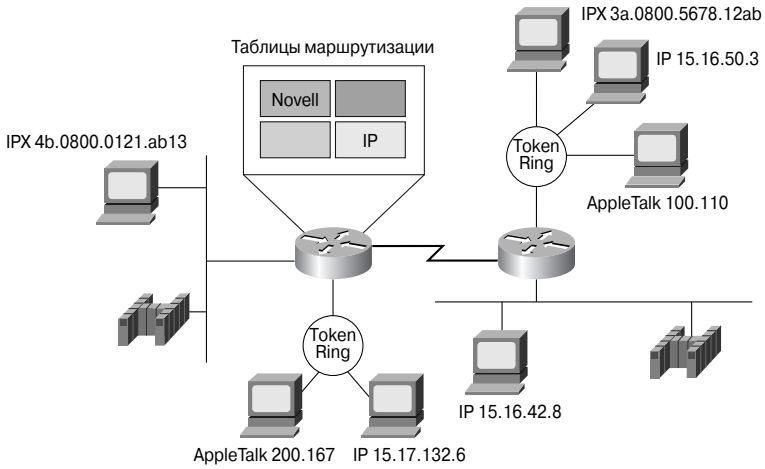


Рис. 10.1. Маршрутизируемые протоколы

- *Протокол маршрутизации* — это протокол, который поддерживает маршрутизируемые протоколы и предоставляет механизмы обмена маршрутной информацией. Сообщения протокола маршрутизации передаются между маршрутизаторами. Протокол маршрутизации позволяет маршрутизаторам обмениваться информацией друг с другом для обновления записей и поддержки таблиц маршрутизации. Ниже приводятся некоторые примеры протоколов маршрутизации TCP/IP:
  - протокол маршрутной информации (Routing Information Protocol — RIP);
  - протокол маршрутизации внутреннего шлюза (Interior Gateway Routing Protocol — IGRP);
  - усовершенствованный протокол маршрутизации внутреннего шлюза (Enhanced Interior Gateway Routing Protocol — EIGRP);
  - протокол первоочередного обнаружения кратчайших маршрутов (Open Shortest Path First — OSPF).

  Легче всего запомнить, что такое протоколы маршрутизации, если представлять себе, что это протоколы обмена маршрутной информацией.

Чтобы протокол был маршрутизируемым, в нем должны наличествовать механизмы назначения как номера сети, так и номера узла для каждого отдельного сетевого устройства. В некоторых протоколах, таких, как, например, IPX, необходимо назначить только адрес сети, поскольку в качестве адреса устройства эта технология использует физический адрес (MAC-адрес) устройства. Другие протоколы, такие, как IP, требуют, чтобы явно был задан весь адрес и сетевая маска.

Для создания маршрутизируемой сети необходимы как *IP-адрес*, так и *маска сети*. Сетевая маска делит 32-битовый IP-адрес на сетевую часть и адрес узла. Протокол IPX использует MAC-адрес, объединенный с установленным администратором номером сети, для создания полного адреса и не требует использования сетевой маски. При использовании IP-технологий адрес сети вычисляется путем сравнения полного адреса и маски подсети.

Сетевая маска позволяет рассматривать группу последовательных IP-адресов как единое целое. Без такой возможности группировки адресов потребовался бы механизм маршрутизации для каждого отдельного узла. Такая схема была бы непригодна для миллионов узлов, работающих в сети Internet. На рис. 10.2 показано, что все 254 адреса в диапазоне от 192.168.10.1 до 192.168.10.254 могут быть представлены одним сетевым адресом 192.168.10.0. Такая возможность позволяет адресовать информацию любому из этих узлов, используя соответствующий адрес сети. Таким образом, таблицы маршрутизации должны содержать всего одну запись — 192.168.10.0 — вместо 254 записей для каждого отдельного узла. Описанный выше подход стандартизован Консорциумом программного обеспечения сети Internet (Internet Software Consorcium — `http://www.isc.org`). Чтобы маршрутизация могла правильно функционировать, рекомендуется использовать группирование адресов.

В последующих разделах описано, как в маршрутизаторах реализованы базовые функции третьего уровня в рамках эталонной модели взаимодействия открытых систем (Open Systems Interconnection — модель OSI). Показано, в чем состоит разница между протоколами маршрутизации и маршрутизируемыми протоколами и реализованными в маршрутизаторах механизмами определения расстояния между удаленными точками. В конце главы подробно описаны технологии маршрутизации по вектору расстояния (distance-vector), состоянию канала (link-state) и гибридная маршрутизация (hybrid), рассказано о том, как каждый тип маршрутизации решает общие проблемы поиска маршрутов и взаимодействия.

Презентация: сравнение маршрутизируемых протоколов и протоколов маршрутизации
В этой видеопрезентации рассматриваются основные различия между протоколами маршрутизации и маршрутизируемыми протоколами. В ней также проиллюстрированы сферы применения протоколов обоих типов.

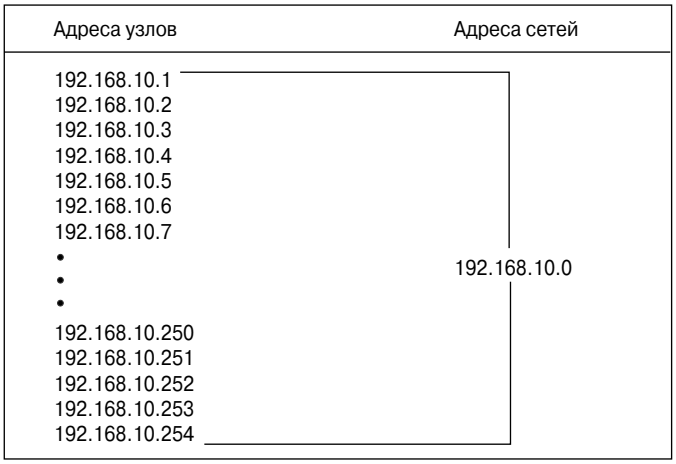


Рис. 10.2. Адреса сетей и узлов

IP как маршрутизируемый протокол

Протокол IP является наиболее широко распространенной реализацией иерархической схемы сетевой адресации. Используемый в сети Internet, протокол IP не отвечает за установку соединений, не является надежным и позволяет реализовать только негарантированную доставку данных. Термин *протокол без установления соединения (connectionless*) означает, что для взаимодействия не требуется выделенный канал, как это происходит во время телефонного звонка, и не существует процедуры вызова перед началом передачи данных между сетевыми узлами. Протокол IP выбирает наиболее эффективный маршрут из числа доступных на основе решения, принятого протоколом маршрутизации. Отсутствие надежности и негарантированная доставка не означает, что система работает плохо и ненадежно, а указывает лишь на то, что протокол IP не предпринимает никаких усилий, чтобы проверить, был ли доставлен пакет по назначению. Эти функции делегированы протоколам верхних уровней.

Информация, проходя сверху вниз по уровням OSI-модели, на каждом из уровней надлежащим образом обрабатывается. На рис. 10.3 показано, что на сетевом уровне данные инкапсулируются внутрь *пакетов*, зачастую называемых *дейтаграммами*.

Протокол IP распознает формат заголовка пакета (включая адресную часть и другую служебную информацию), но никоим образом не заботится о фактических данных. Он принимает любые данные, переданные протоколами верхнего уровня (рис.10.4).

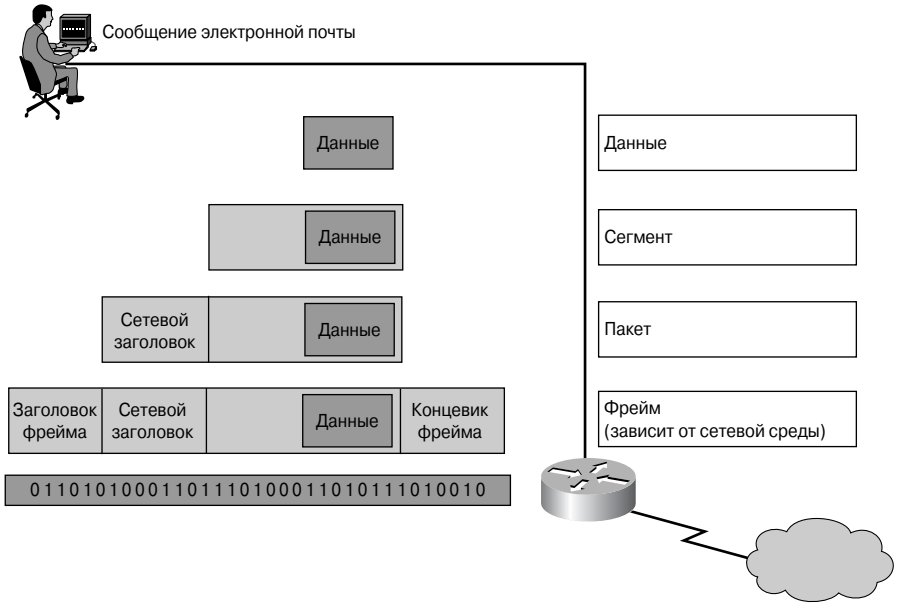


Рис. 10.3. Инкапсуляция

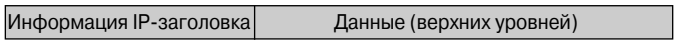


Рис. 10.4. IP-заголовок

Пересылка пакетов и коммутация внутри маршрутизатора

На рис. 10.5 показано, что заголовок и концевик фрейма отбрасываются и заменяются новыми каждый раз при прохождении движущимся по сети пакетом маршрутизирующего устройства третьего уровня. Причина этого состоит в том, что блоки информации второго уровня (фреймы) используются для локальной доставки информации, в то время как блоки третьего уровня (пакеты) предназначены для сквозной передачи данных согласно схеме адресации.

Ethernet-фреймы второго уровня предназначены для работы внутри широковещательных доменов с назначенными каждому сетевому устройству MAC-адресами. Фреймы второго уровня других типов, такие, как последовательные двухточечные соединения и Frame Relay распределенных сетей (сетей WAN), используют свою собственную схему адресации второго уровня. Принципиальным является то, что, независимо от используемой схемы адресации второго уровня, все они разработаны для использования внутри одного широковещательного домена второго уровня. При прохождении данными через устройство третьего уровня информация второго уровня изменяется.

Процессы, выполняемые устройствами третьего уровня, проиллюстрированы на рис. 10.6 и описаны в следующем абзаце.

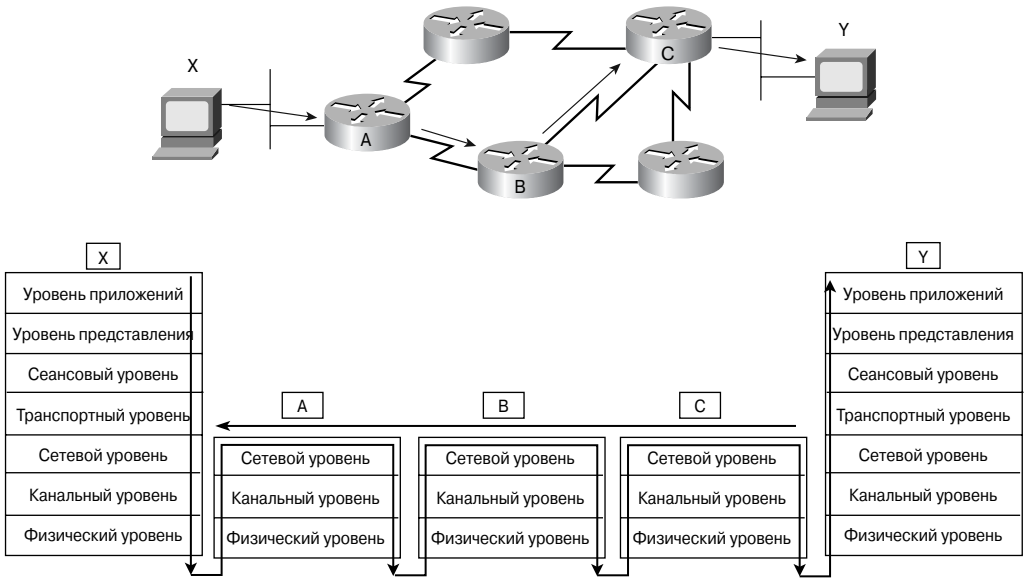


Рис. 10.5. Поток данных сетевого уровня

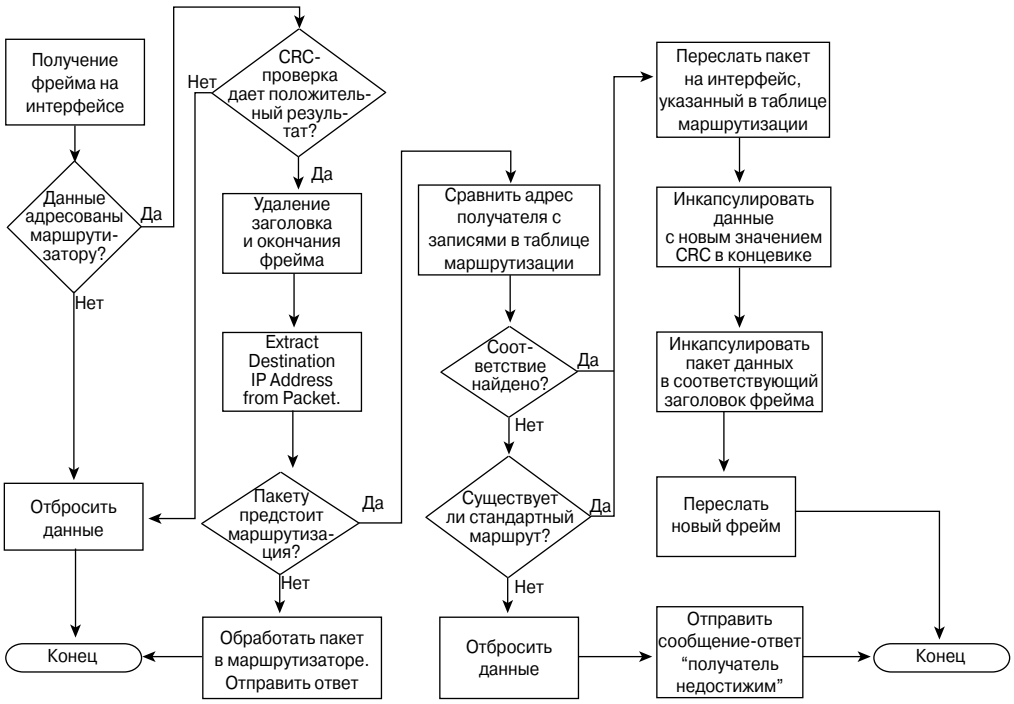


Рис. 10.6. Изменение пакета в процессе инкапсуляции в маршрутизаторе

Из пакета, приходящего на интерфейс маршрутизатора, извлекается MAC-адрес и проверяется, адресован ли этот пакет непосредственно какому-либо узлу либо интерфейсу или он является широковещательным; та же процедура выполняется всеми устройствами внутри *домена коллизий*. В любом из указанных вариантов пакет будет принят и обработан; в противном случае пакет будет отброшен, поскольку был адресован другому устройству в домене коллизий. Таким образом, домен коллизий — это разделяемая среда передачи данных, в которой устройства работают в режиме конкуренции. На основании значения, содержащегося в поле контрольной суммы, с помощью циклического избыточного кода (Cyclical Redundancy Check — CRC), извлеченного из окончания полученного фрейма, проверяется, не были ли данные повреждены. Если проверка дает отрицательный результат, такой фрейм отбрасывается. В случае положительного результата проверки заголовок и окончание фрейма удаляются, и пакет передается на третий уровень. Далее выполняется проверка, адресован ли пакет маршрутизатору или потребуется его дальнейшая маршрутизация на пути к пункту назначения. Пакеты, адресованные маршрутизатору в качестве IP-адреса получателя, содержат адрес одного из интерфейсов маршрутизатора. У таких пакетов удаляется заголовок, и они передаются на четвертый уровень. Если пакету предстоит маршрутизация, он сравнивается с записями в таблице маршрутизации. Если будет найдено точное соответствие или существует стандартный маршрут, пакет будет отправлен на интерфейс, указанный в соответствующей записи таблицы маршрутизации.

Когда пакет коммутируется на выходной интерфейс, новое значение CRC добавляется в конец фрейма и, в зависимости от типа интерфейса (Ethernet, Frame Relay или последовательный), пакету добавляется соответствующий заголовок. После этого фрейм пересылается в другой широковещательный домен, являющийся следующей частью маршрута к конечному пункту назначения.

Сетевые службы с установлением соединения и без

Большинство сетевых служб модели OSI используют системы доставки без установления соединения (протокол UDP), как это показано на рис. 10.7. Они работают с каждым пакетом в отдельности и пересылают их в нужном направлении через сеть. Пакеты могут быть переданы по сетевым маршрутам и будут собраны вместе в сообщение только тогда, когда достигнут своего пункта назначения. В системах без установления соединения перед отправкой пакета контакт с получателем не происходит. Хорошей аналогией систем без установления соединения может быть традиционная почтовая служба. Никто не связывается с получателем, перед тем как отправить ему письмо. Оно отправляется по некоторому маршруту, и получатель узнает о нем только в тот момент, когда получает письмо.

Сетевые службы без установления соединения часто называют процессами с *коммутацией пакетов*. В таких процессах пакеты могут проходить разными маршрутами от отправителя к получателю, а также (что вполне вероятно) прибывать в пункт назначения в другом порядке. Устройства выбирают маршрут для каждого пакета на основе различных критериев. Некоторые критерии (как, например, доступная полоса пропускания) могут быть разными для различных пакетов.

Сеть Internet — огромная объединенная сеть без установления соединения, в которой за доставку всех пакетов отвечает протокол IP. Протокол TCP (четвертый уровень модели) использует службы с установлением соединений поверх протокола IP (третьего уровня). Сегменты TCP инкапсулируются в IP-пакеты для передачи через сеть Internet.

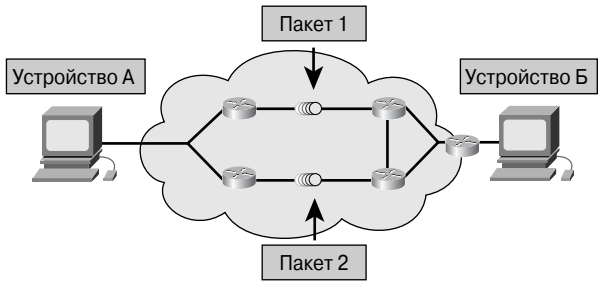


Рис. 10.7. Система доставки без установления соединения для сетевых служб

Протокол IP является системой без установления соединений: каждый пакет в ней обрабатывается отдельно. Например, при использовании FTP-клиента для передачи файлов протокол IP не посылает данные единым неразрывным потоком; он работает с каждым пакетом отдельно. Каждый пакет может пойти своим маршрутом; некоторые из них даже могут быть утеряны. Протокол IP полагается на протоколы транспортного уровня, чтобы определить, был ли доставлен пакет, и при необходимости организовать повторную передачу. Транспортный уровень также отвечает за сборку пакетов в сообщение в надлежащей последовательности.

В системах с *установлением соединения*, как следует из названия, соединение между отправителем и получателем устанавливается до начала передачи данных, как это показано на рис. 10.8. Примером сети с установлением соединения является телефонная система. Взаимодействие начинается только после того, как будет произведен звонок и установлено соединение. В сетях с установлением соединения вначале организуется соединение с получателем и только после этого начинается фактическая передача данных. Все пакеты передаются последовательно, с использованием одного и того же физического канала или, в самом общем случае, одного виртуального канала.

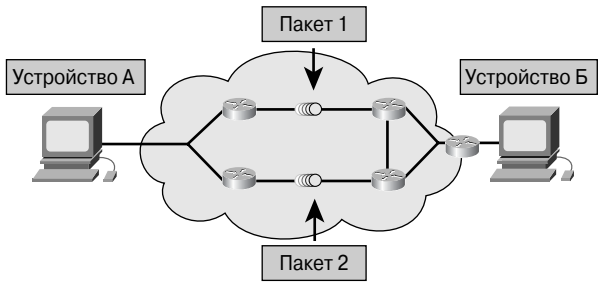


Рис. 10.8. Система доставки с установлением соединения для сетевых служб

Структура IP-пакета

Ранее было рассмотрено, как пакеты или дейтаграммы третьего уровня становятся данными второго уровня и инкапсулируются во фреймы.

Аналогично, как показано на рис. 10.9, IP-пакеты состоят из данных верхнего уровня и IP-заголовка.

- **Версия (Version)** — четырехбитовое поле, описывающее используемую версию протокола IP. Все устройства обязаны использовать протокол IP одной версии; устройство, использующее другую версию, будет отбрасывать пакеты.
- **Длина IP-заголовка (IP Header Length — HLEN)** — четырехбитовое поле, описывающее длину заголовка дейтаграммы в 32-битовых блоках. Данное значение — это полная длина заголовка с учетом двух полей переменной длины.
- **Тип обслуживания (Type of Service — TOS)** — восьмибитовое поле, указывающее на степень важности информации, которая присвоена определенным протоколом верхнего уровня.
- **Полная длина (Total Length)** — шестнадцатибитовое поле, описывающее полную длину пакета в байтах, включая данные и заголовок. Чтобы вычислить длину блока данных, нужно из полной длины вычесть значение поля HLEN.
- **Идентификация (Identification)** — шестнадцатибитовое поле, хранящее целое число, описывающее данную дейтаграмму. Это число представляет собой последовательный номер.
- **Флаги (Flags)** — трехбитовое поле, в котором два младших бита контролируют фрагментацию пакетов. Первый бит определяет, был ли пакет фрагментирован, а второй — является ли этот пакет последним фрагментом в серии фрагментированных пакетов.
- **Смещение фрагментации (Fragment Offset)** — тринадцатибитовое поле, помогающее собрать вместе фрагменты дейтаграммы. Это поле позволяет использовать 16 битов для поля флагов.
- **Время жизни (Time-to-Live — TTL)** — восьмибитовое поле, в котором хранится последовательно уменьшающееся значение счетчика, вплоть до нуля. В последнем случае (счетчик равен нулю) дейтаграмма будет отброшена — таким образом предотвращается бесконечная циклическая пересылка пакета. Аналогом этого поля является счетчик узлов в протоколах маршрутизации.
- **Протокол (Protocol)** — восьмибитовое поле, указывающее, какой протокол верхнего уровня получит пакет, после того как обработка протоколом IP будет закончена. Примерами значений в этом поле являются протоколы TCP и UDP.
- **Контрольная сумма заголовка (Header Checksum)** — шестнадцатибитовое поле, которое помогает проверить целостность заголовка пакета.
- **IP-адрес отправителя (Source IP address)** — 32-битовое поле, содержащее IP-адрес узла-отправителя.

- **IP-адрес получателя (Destination IP address) —** 32-битовое поле, содержащее IP-адрес узла-получателя.
- **Опции (Options) —** поле переменной длинны, позволяющее протоколу IP реализовать поддержку различных опций, например, средств безопасности.
- **Дополнение (Padding) —** поле, используемое для вставки дополнительных нулей, чтобы гарантировать кратность IP-заголовка 32 битам.
- **Данные (Data)** — поле переменной длины (максимум 64 Кбит), содержащее информацию верхних уровней.

Рис. 10.9. Структура IP-пакета

IP-пакет состоит из данных протокола верхнего уровня и заголовка, который имеет описанную выше структуру. Хотя до сих пор основное внимание в этой книге уделялось IP-адресам отправителя и получателя, именно другие части IP-заголовка делают его столь гибким и надежным. Информация, хранящаяся в полях заголовка, задает данные пакета и предназначена для протоколов верхних уровней. Выше, в нескольких предыдущих главах, обсуждалась идея о независимости уровней; информация заголовка — это механизм, реализующий такую независимость.

Протоколы IP-маршрутизации

Основным камнем преткновения для тех, кто только начинает изучать сетевые технологии, является отличие маршрутизируемых протоколов от протоколов маршрутизации. Два термина звучат очень похоже[1], тем не менее, они обозначают принципиально разные понятия. В следующем разделе основное внимание уделено протоколам маршрутизации, которые отвечают за построение таблиц маршрутизации маршрутизаторами и поиск оптимального маршрута к узлу в сети Internet.

[1] *Чаще всего два термина путают в английском написании:* routing *и* routed. *— Прим. ред.*

Обзор технологии маршрутизации

Маршрутизация является функцией третьего уровня модели OSI. Она основана на иерархической схеме, которая позволяет группировать отдельные адреса и работать с группами как с единым целым до тех пор, пока не потребуется установить индивидуальный адрес для окончательной доставки данных. Под термином "маршрутизация" подразумевают процесс определения наиболее эффективного пути от одного устройства к другому (рис. 10.10). Основным устройством, отвечающим за осуществление процесса маршрутизации, является *маршрутизатор.*

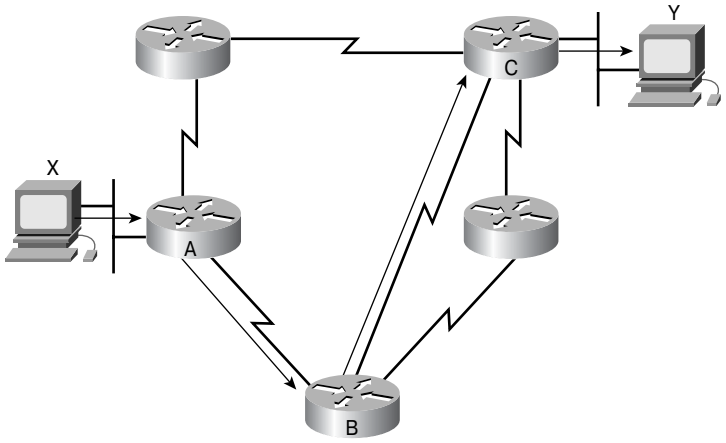


Рис. 10.10. Принцип работы протокола сетевого уровня

Маршрутизатор выполняет две ключевые функции:

- поддерживает таблицы маршрутизации и обменивается информацией об изменениях в топологии сети с другими маршрутизаторами. Эта функция реализуется с помощью одного или нескольких протоколов маршрутизации для передачи сетевой информации другим маршрутизаторам;
- когда пакеты приходят на один из интерфейсов, маршрутизатор, руководствуясь таблицей маршрутизации, должен определить, куда именно следует отправить пакет. Он перенаправляет пакеты на выбранный интерфейс, создает фреймы и затем пересылает их.

Маршрутизатор является устройством сетевого уровня и использует одну или несколько *метрик маршрутизации (routing metric*), для того чтобы установить оптимальный путь, по которому должен следовать сетевой трафик. Метрика маршрутизации — это параметр, по которому определяется наиболее предпочтительный маршрут. На рис. 10.11 показано, что протоколы маршрутизации используют различные комбинации параметров для расчета метрик.

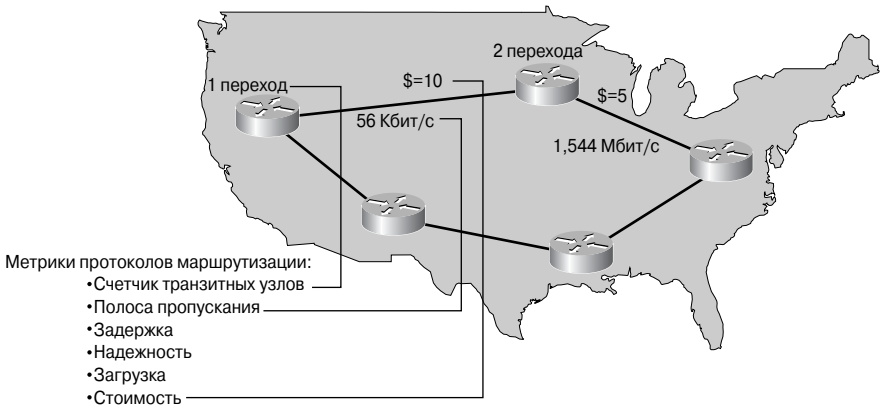


Рис. 10.11. Метрики протокола маршрутизации

Для определения наилучшего межсетевого маршрута вычисляются различные комбинации компонентов метрики: количество ретрансляций (т.е. транзитных узлов), полоса пропускания, задержки, надежность, загрузка и стоимость. Маршрутизаторы объединяют сетевые сегменты или целые сети. Фреймы данных они передают на основе информации протокола третьего уровня. Маршрутизаторы принимают логическое решение о наилучшем маршруте доставки данных между сетями и отправляют пакеты в соответствующий исходящий порт для последующей инкапсуляции и пересылки. Процессы инкапсуляции и декапсуляции происходят каждый раз, когда пакеты проходят через маршрутизатор и данные передаются от одного устройства другому (рис. 10.12). При выполнении инкапсуляции поток данных разбивается на сегменты, добавляются необходимые заголовки и концевики, после чего данные передаются по сети. Декапсуляция — это обратный процесс, при котором удаляются заголовки и концевики, а данные собираются в неразрывный поток. Маршрутизаторы принимают фреймы от устройств локальной сети (например, рабочих станций) и на основе информации третьего уровня пересылают их по сети.

Эта глава и остальная часть книги посвящены наиболее широко используемому маршрутизируемому протоколу — IP. Несмотря на то что далее обсуждается только протокол IP, следует знать, что существуют другие маршрутизируемые протоколы, такие, как IPX/SPX и AppleTalk.

В протоколах IPX/SPX и AppleTalk реализована поддержка средств третьего уровня, благодаря чему они могут маршрутизироваться. Протоколы, не поддерживающие третий уровень, называются немаршрутизируемыми. Наиболее распространенным из их числа является транспортный протокол, используемый всеми сетевыми ОС фирмы Microsoft (NetBIOS Extended User Interface — NetBEUI) — простой и эффективный протокол, область использования которого ограничена одним сегментом сети.

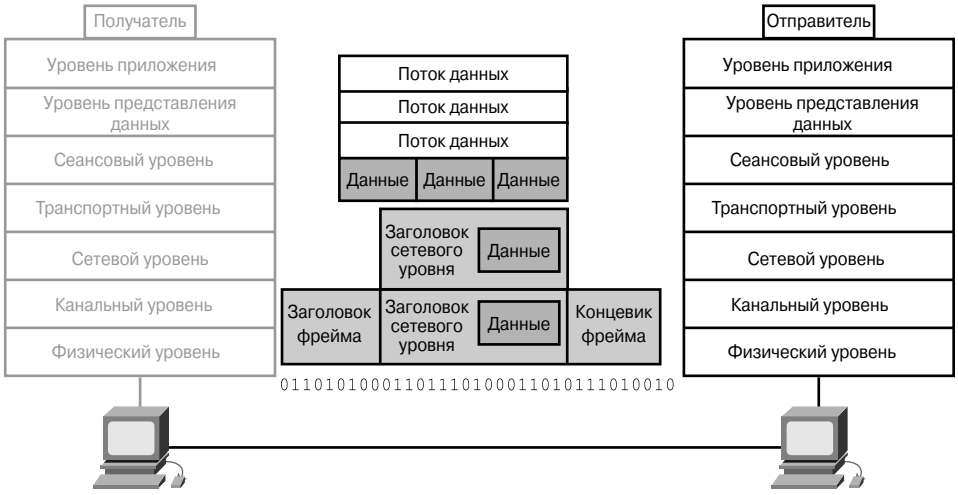


Рис. 10.12. Инкапсуляция данных

Сравнение маршрутизации и коммутации

Маршрутизацию часто путают с коммутацией второго уровня, которая, как может показаться при поверхностном рассмотрении, выполняет те же функции. Принципиальное различие состоит в том, что коммутация реализована на втором уровне модели OSI, а маршрутизация — на третьем. Такое принципиальное отличие означает, что маршрутизация и коммутация используют разную информацию для организации передачи данных от отправителя получателю.

Как коммутация соотносится с маршрутизацией, можно пояснить на примере местных и междугородних телефонных звонков. Для обслуживания местного телефонного звонка (с тем же кодом региона) используется местная телефонная станция. Понятно, что местная АТС хранит только местные номера и ничего не знает о телефонных номерах абонентов из других регионов. При получении звонка, номер которого находится вне компетенции местной станции, она коммутирует такой звонок станции более высокого уровня, которая хранит коды регионов. Станция более высокого уровня коммутирует звонок таким образом, что в конце концов он будет получен местной станцией, обслуживающей номера с кодом региона, по которому был сделан звонок.

Как показано на рис. 10.13, маршрутизатор выполняет функции, подобные тем, которые осуществляет телефонная станция высокого уровня в телефонной сети. Когда говорят о коммутации второго уровня, применяемой в локальных сетях, ее часто связывают с таким понятием, как *широковещательный домен (broadcast domain*). Маршрутизация третьего уровня предназначена для передачи данных между широковещательными доменами и требует иерархической схемы адресации, что и реализовано в протоколах третьего уровня, как, например, в протоколе IP. Коммутатор второго уровня ничего не знает об IP-адресах и может работать только с локальными MAC-адресами узлов. Когда узел отправляет информацию нелокальному получателю,

он адресует фрейм своему стандартному шлюзу-маршрутизатору, используя для этого MAC-адрес маршрутизатора.

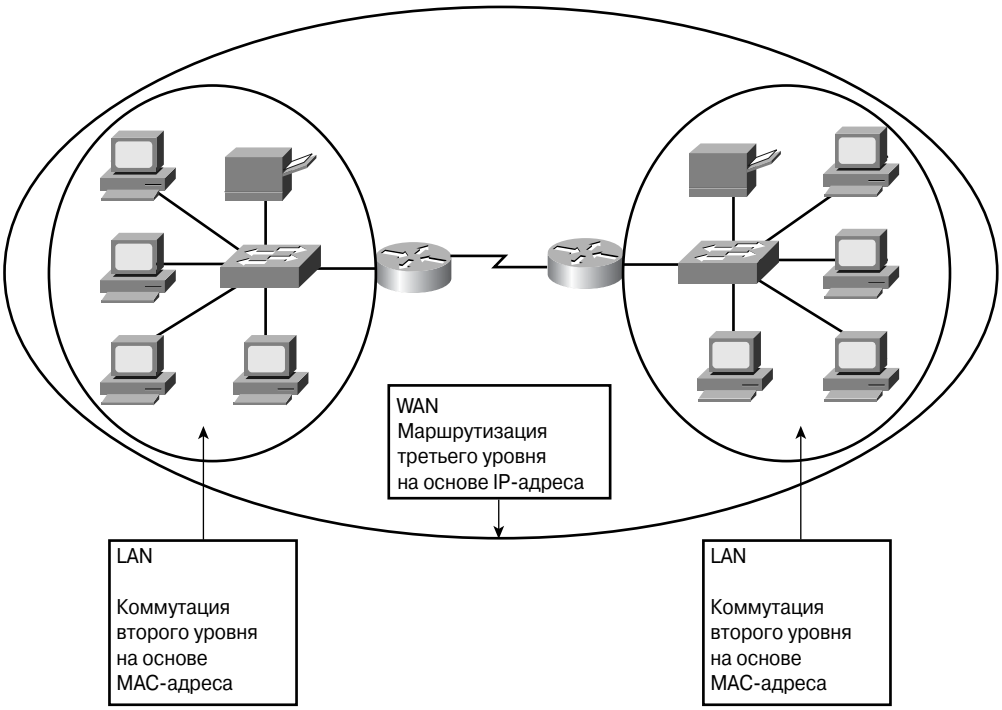


Рис. 10.13. Коммутация второго уровня и маршрутизация третьего

Коммутатор второго уровня объединяет сегменты, принадлежащие одной логической сети или *подсети (subnetwork*). Если узлу X необходимо переслать фрейм получателю из другой сети или подсети, он отправляет фрейм маршрутизатору, который тоже подключен к коммутатору. Узел X знает IP-адрес маршрутизатора, поскольку в его конфигурации протокола IP указан IP-адрес стандартного шлюза, но он ничего не знает о MAC-адресе шлюза. Используя протокол преобразования адресов (Address Resolution Protocol — ARP), который переводит IP-адреса в MAC-адреса, узел X выясняет MAC-адрес маршрутизатора. Коммутатор передает фрейм маршрутизатору на основе его MAC-адреса. Маршрутизатор анализирует адрес получателя третьего уровня в пакете для принятия решения о выборе маршрута. Стандартный шлюз — это маршрутизатор, находящийся в той же сети или подсети, что и узел X. Подобно тому, как коммутатор второго уровня хранит таблицу известных MAC-адресов, маршрутизатор работает с набором IP-адресов сетей, который формирует базу данных доступных ему сетей, называющуюся таблицей маршрутизации (рис. 10.14).

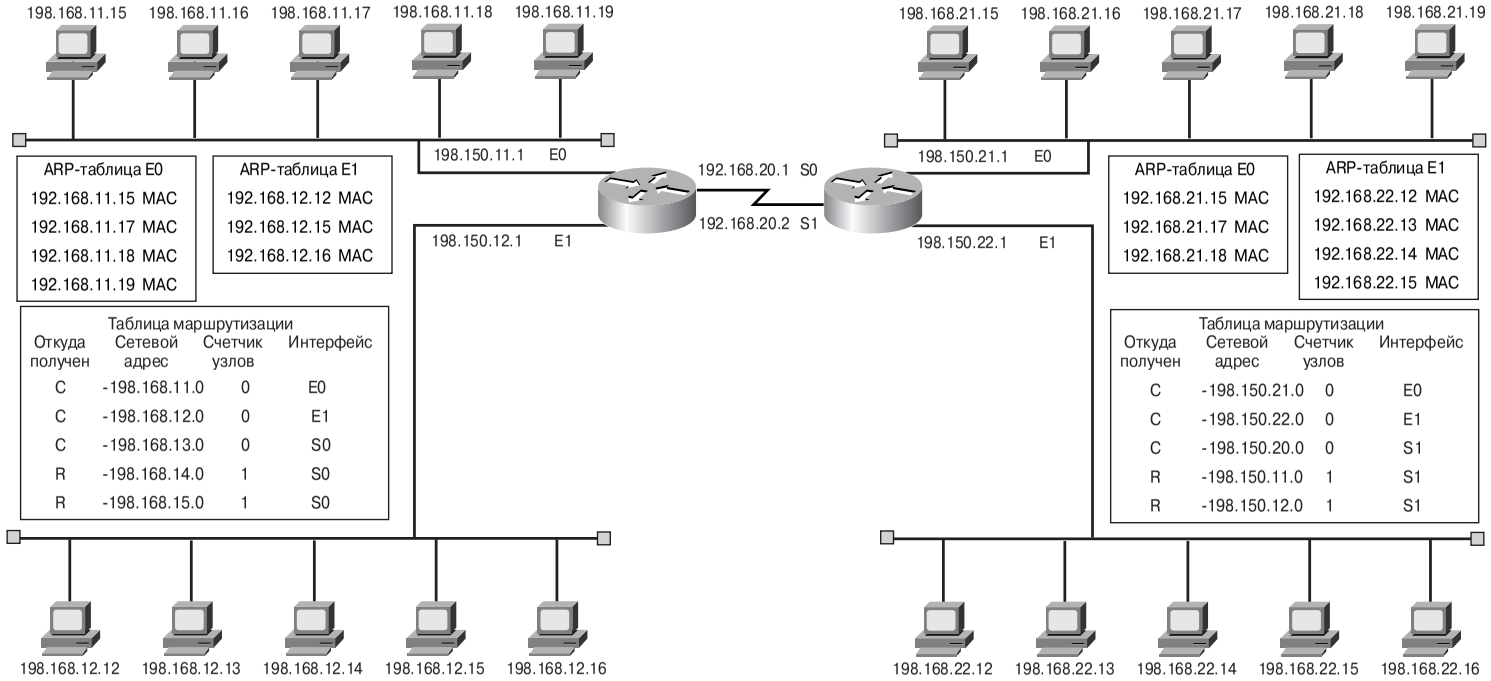


Рис. 10.14. Таблицы маршрутизации и ARP-таблицы маршрутизатора

Каждый компьютер и Ethernet-интерфейс маршрутизатора поддерживают ARP-таблицу для взаимодействий второго уровня; такие таблицы актуальны только для того широковещательного домена, к которому подключено данное устройство. Маршрутизатор, кроме этого, поддерживает еще и таблицу маршрутизации, которая дает возможность выбирать маршрут для доставки данных за пределы широковещательного домена. Каждая ARP-таблица содержит пары IP- и MAC-адресов. (На рис. 10.14 для краткости MAC-адреса представлены аббревиатурой MAC, поскольку фактические их значения имеют слишком длинную запись и не поместились бы на рисунке). *Таблица маршрутизации* содержит информацию о маршрутах; в данном случае — признак: непосредственно подключенная сеть (обозначена символом "C") и сеть, которая получена по протоколу RIP (обозначена символом "R"), IP-адреса доступных сетей, значение счетчика транзитных узлов до этих известных сетей и интерфейсы, через которые информация будет отправлена в нужную сеть. Разница между двумя рассмотренными типами адресов состоит в том, что MAC-адреса не организованы по какому-то определенному принципу. Однако этот недостаток не вызывает проблем с управлением сетями, поскольку отдельные сетевые сегменты не содержат большого количества узлов. Если бы IP-адреса подчинялись тем же правилам, сеть Internet просто не смогла бы функционировать. В том случае, если бы IP-адреса не были организованы (иерархически или как-либо еще), то не существовало бы способа определить маршрут для достижения каждого конкретного адреса. Иерархическая организация IP-адресов позволяет рассматривать группы адресов как единое целое до тех пор, пока не потребуется определить адрес индивидуального узла. Понять такой подход в адресации можно на примере библиотеки, хранящей миллионы отдельных страниц в одной большой кипе бумаг. В таком случае воспользоваться необходимым материалом будет невозможно, поскольку нет способа найти необходимый документ. Намного проще воспользоваться нужной информацией, если страницы пронумерованы, переплетены в книги и каждая внесена в каталог.

Еще одно отличие между коммутируемыми и маршрутизируемыми сетями заключается в том, что коммутируемые сети второго уровня не блокируют широковещательные рассылки третьего уровня. Вследствие этого они могут быть подвержены широковещательным штормам. Маршрутизаторы обычно блокируют широковещательные пакеты, ограничивая таким образом зону действия широковещательных штормов локальным широковещательным доменом. Дополнительно благодаря блокировке широковещательных рассылок маршрутизаторы предоставляют более высокий, чем коммутаторы, уровень защиты и контроль полосы пропускания.

Функции маршрутизации и коммутации сравниваются в табл. 10.1.

Интерактивная презентация: сравнение коммутации и маршрутизации
В этой презентации рассмотрены различия функций маршрутизации и коммутации.

Таблица 10.1. Сравнение функций маршрутизатора и коммутатора

Функция	Маршрутизатор	Коммутатор
Скорость	Медленнее	Быстрее
Уровень OSI	Уровень 3	Уровень 2
Используемая адресация	IP	MAC
Широковещательные рассылки	Блокируются	Пропускаются
Безопасность	Выше	Ниже
Сегментация сетей	Сегментирует сеть на широковещательные домены	Сегментирует сеть на домены коллизий

Сравнение маршрутизируемых протоколов и протоколов маршрутизации

Протоколы сетевого уровня делятся на две категории: маршрутизируемые и протоколы маршрутизации (рис. 10.15). Маршрутизируемые протоколы организуют передачу данных через сеть, а протоколы маршрутизации реализуют механизмы, с помощью которых маршрутизаторы определяют необходимое направление для доставки данных из одного пункта в другой.

Протоколы, способные передавать данные от одного узла другому, находящемуся за маршрутизатором, называются маршрутизируемыми, или просто протоколами передачи данных.

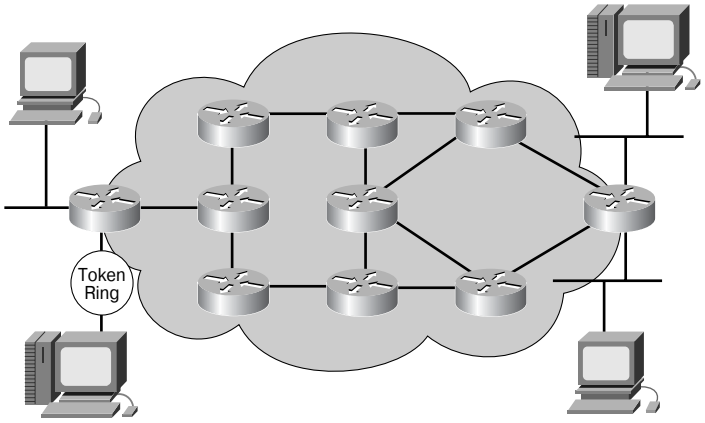
Принцип работы маршрутизируемого протокола проиллюстрирован на рис. 10.16.

- Маршрутизируемым является любой протокол или набор сетевых протоколов, которые предоставляют маршрутизаторам необходимую информацию в адресе сетевого уровня для передачи данных следующему узлу и конечному получателю.
- Маршрутизируемый протокол задает формат пакета и использование в нем отдельных полей. В большинстве своем пакеты передаются от одной конечной системы другой.

Примерами маршрутизируемых протоколов являются IP и IPX. Кроме того, примерами таких протоколов могут служить DECnet, AppleTalk, Banyan VINES и Xerox Networ System (XNS), но следует помнить, что эти протоколы уже устарели и на практике встречаются достаточно редко.

Маршрутизаторы используют протоколы маршрутизации для обмена таблицами маршрутизации и совместного использования информации о доступных маршрутах. Иными словами, *протоколы маршрутизации* дают возможность маршрутизаторам выбирать маршрут для *маршрутизируемых протоколов*, после того как будут обнаружены все возможные пути к получателю.

Маршрутизируемые протоколы используются для передачи трафика пользователя. Примерами таких протоколов являются IP и IPX.



Протоколы маршрутизации используются для обмена информацией и таблицами маршрутизации между маршрутизаторами. Примерами таких протоколов могут служить RIP, IGRP и OSPF.

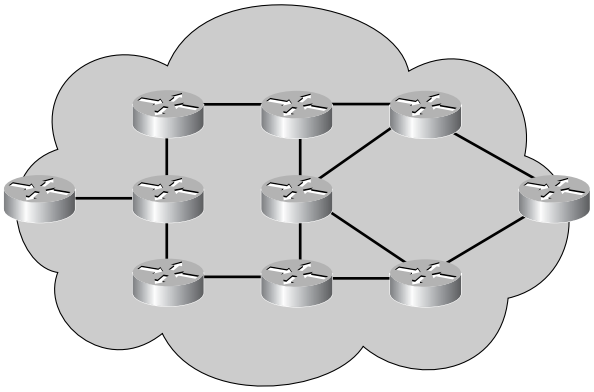


Рис. 10.15. Маршрутизируемый протокол и протокол маршрутизации

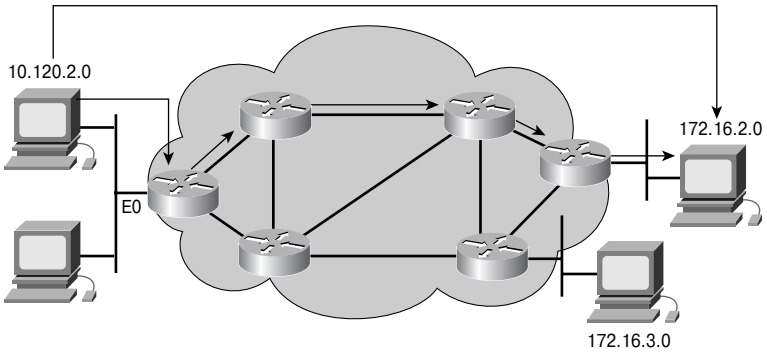
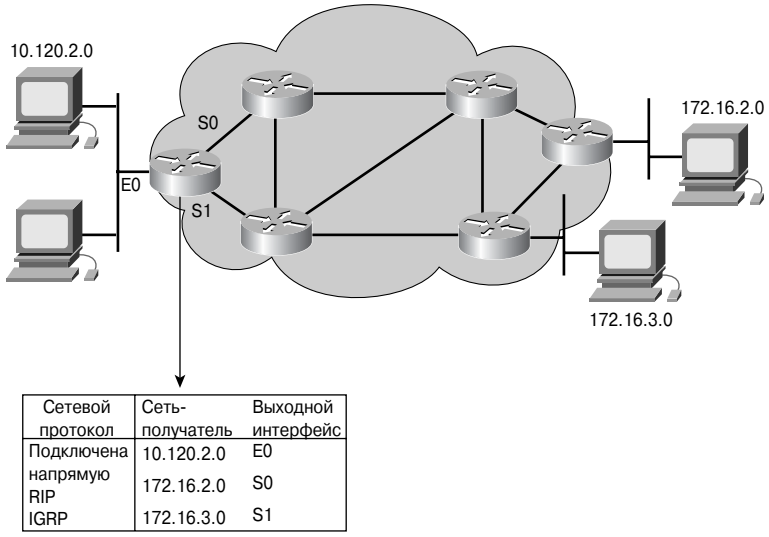


Рис. 10.16. Маршрутизируемый протокол

Принцип работы протокола маршрутизации проиллюстрирован на рис. 10.17.

- Протокол маршрутизации обеспечивает работу процесса совместного использования информации о доступных маршрутах.
- Протокол маршрутизации позволяет маршрутизаторам обмениваться информацией друг с другом для поддержки таблиц маршрутизации.

Примерами протоколов маршрутизации, которые поддерживают маршрутизируемый протокол IP, являются RIP, IGRP, OSPF, протокол граничного шлюза (Border Gateway Protocol — BGP) и EIGRP.



Сетевой протокол	Сеть-получатель	Выходной интерфейс
Подключена напрямую	10.120.2.0	E0
	172.16.2.0	S0
RIP	172.16.3.0	S1
IGRP		

Рис. 10.17. Протокол маршрутизации

Поиск оптимального маршрута

Процесс нахождения оптимального пути, по которому следует передать пакет, выполняется на третьем уровне эталонной модели OSI (сетевом). Эта процедура позволяет маршрутизатору оценить существующие маршруты к получателю и выбрать среди них наиболее предпочтительный. Как показано на рис. 10.18, службы маршрутизации используют информацию о топологии сети в процессе анализа сетевых маршрутов. Определением пути называют процесс, используемый маршрутизатором для выбора следующего узла на пути следования пакета к своему конечному пункту назначения. Этот процесс также называется *маршрутизацией* пакета.

Процесс поиска маршрута для пакета можно сравнить с поездкой из одной части города в другую. У водителя есть карта, где указаны улицы, выбирая которые, он движется к пункту назначения. Отрезок от одного перекрестка до другого аналогичен прохождению пакетом расстояния между двумя маршрутизаторами, который обычно называют *транзитным переходом*. Похожим образом маршрутизатор использует

"карту", на которой показано наличие доступных путей до пункта назначения. Маршрутизаторы могут принимать решения, основываясь на информации об интенсивности трафика и пропускной способности соединения, так же, как водитель может выбрать более быстрый путь (скоростное шоссе) или ехать по менее загруженным обходным улицам. В этом разделе показано, как маршрутизатор выбирает наилучший путь для пакета, следующего из одной сети в другую.

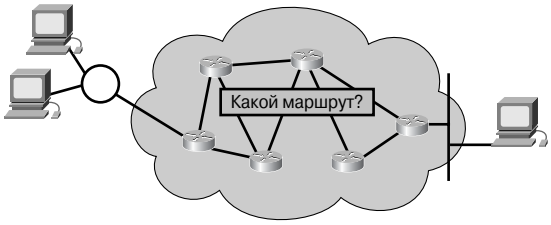


Рис. 10.18. Выбор маршрута

Решение, которое принимает водитель, определяется многими факторами: загруженностью дорог, количеством полос на дороге, стоимостью проезда по магистрали и тем, как часто дорога бывает закрыта. Иногда бывает значительно выгоднее проехать по более длинному маршруту, т.е. по более узкой, но менее загруженной дороге вместо широкой магистрали, по которой ездит огромное количество машин и где часто бывают пробки. Аналогично маршрутизаторы принимают решение о выборе оптимального пути на основании загрузки, полосы пропускания, задержки, стоимости и надежности какого-либо канала. Процесс выбора маршрута для каждого пакета включает в себя следующие компоненты:

- адрес получателя берется непосредственно из заголовка пакета;
- сетевая маска первой записи в таблице маршрутизации применяется к адресу получателя в пакете;
- после того как маска умножается на адрес получателя (логическая операция "И"), полученная величина сравнивается с записью в таблице маршрутизации;
- если оба значения совпали, пакет пересылается на интерфейс (порт) маршрутизатора, с которым связана данная запись в таблице маршрутизации;
- если же совпадений значений нет, описанным выше образом проверяется следующая запись в таблице маршрутизации;
- если адрес пакета не соответствует ни одной из записей в таблице маршрутизации, маршрутизатор проверяет, есть ли у него стандартный маршрут;
- если в маршрутизаторе сконфигурирован стандартный маршрут, пакет передается на соответствующий ему порт маршрутизатора. *Стандартный маршрут* (default route) — это маршрут, который конфигурирует в устройстве системный администратор и который будет использоваться устройством в том случае, если не найдены соответствия ни одной записи в таблице маршрутизации;

- если же стандартного маршрута нет, то пакет будет отброшен маршрутизатором. Зачастую в обратном направлении устройство отправляет сообщение, которое сигнализирует о том, что сеть получателя недоступна.

Дополнительная информация: функции сетевого уровня

Адресация сетевого уровня

Сетевой адрес помогает маршрутизатору находить путь в межсетевой среде, а также предоставляет иерархическую информацию или сведения о подсетях. Маршрутизатор использует сетевой адрес для поиска сети-получателя, в которую следует пакет. В дополнение к сетевому адресу сетевой протокол также использует некую форму адреса узла. Для некоторых протоколов сетевого уровня сетевой администратор назначает адреса узлов на основе заранее заданных правил сетевой адресации. Для других протоколов сетевого уровня назначение сетевых адресов происходит частично или полностью динамически, либо автоматически. На рис. 10.19 показаны три устройства в сети 1 (две станции и один маршрутизатор), каждое из которых имеет свой уникальный адрес. (На рисунке также показано, что маршрутизатор подключен к двум другим сетям с номерами 2 и 3.)

Сеть	Узел
1	1 2 3
2	1
3	1

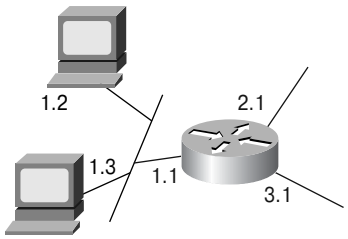


Рис. 10.19. Адреса сетей

Логическая адресация происходит на сетевом уровне. Вспомним аналогию между сетевыми адресами и телефонными номерами. Первая часть телефонного адреса представляет собой код региона, несколько первых цифр номера (в крупных городах — обычно три цифры) указывают на телефонную станцию. Последние цифры номера указывают оборудованию телефонной компании, на какой конкретный телефонный аппарат необходимо направить звонок. Последняя часть телефонного номера похожа на часть адреса, которая описывает узел. Часть целого адреса, содержащая адрес узла, дает маршрутизатору информацию о конкретном устройстве, которому необходимо доставить пакет.

Маршрутизация не может быть реализована без адресации сетевого уровня. Маршрутизаторам необходимо знать сетевые адреса, чтобы обеспечить надлежащую доставку пакетов.
Без иерархической структуры адресации пакеты было бы невозможно передавать между сетями.

Аналогично без определенной иерархии в схеме телефонных номеров, почтовых адресов и транспортных систем не существовало бы надежной доставки товаров и услуг.

MAC-адрес можно сравнить с именем получателя, а адрес сетевого уровня — с почтовым адресом (сетевой адрес и адрес узла). Так, например, если адресат переехал в другой город, имя его останется без изменений, но почтовый адрес должен отражать его новое местоположение. Сетевые устройства (отдельные компьютеры и маршрутизаторы) имеют как MAC-адрес, так и адрес сетевого уровня. Перемещение компьютера в другую сеть не изменит его MAC-адрес, но обязательно потребует назначения нового адреса сетевого уровня.

Коммуникационный путь

Задача сетевого уровня состоит в поиске наилучшего маршрута через сеть. Говоря практическим языком, устройства сети постоянно распространяют определенный набор доступных маршрутов между маршрутизаторами. На рис. 10.20 каждая линия, соединяющая маршрутизаторы, имеет номер, используемый маршрутизатором в качестве сетевого адреса. Эти адреса должны отражать информацию, использующуюся в процессе маршрутизации. Это означает, что адрес должен нести информацию о маршруте для данного физического соединения, которая необходима для передачи пакетов от отправителя получателю в процессе маршрутизации.

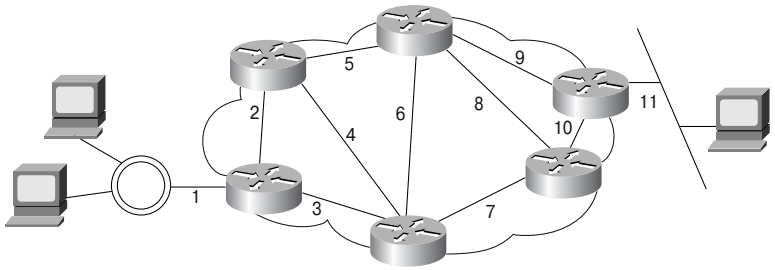


Рис. 10.20. Сетевые соединения

Благодаря использованию иерархических адресов на сетевом уровне создаются соединения, которые позволяют обеспечить взаимодействие между независимыми сетями. Логичность и связность адресов третьего уровня во всей сети улучшают эффективность использования пропускной способности, поскольку устраняют необходимость в ненужных широковещательных запросах. *Широковещательные рассылки* приводят к нежелательной загрузке и бесполезному расходу ресурсов устройств и каналов связи, которые не нуждаются в получении широковещательных пакетов. Использование сквозной схемы адресации для описания путей физических соединений дает возможность сетевому уровню находить путь к получателю, не прибегая к нежелательным перегрузкам устройств и каналов связи из-за использования широковещательных рассылок.

Таблицы маршрутизации

Чтобы найти маршрут, по которому следует передавать данные, протоколы маршрутизации создают и поддерживают таблицы маршрутизации (рис. 10.21). Информация о маршруте может отличаться в зависимости от используемого протокола маршрутизации. Таблица маршрутизации заполняется соответствующим протоколом различной информацией.

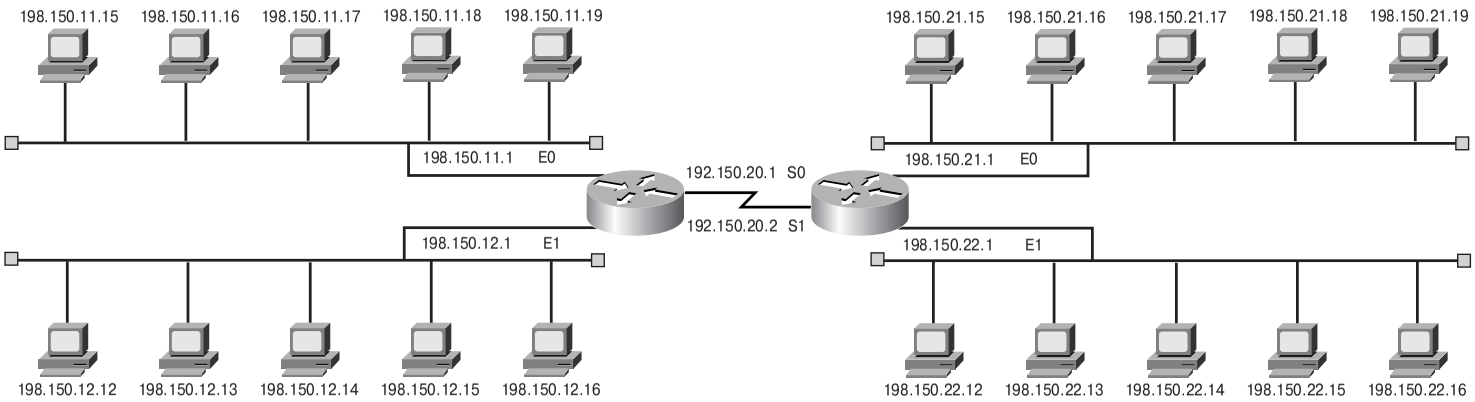


Таблица маршрутизации

Откуда получен	Сетевой адрес	Счетчик узлов	Интерфейс
C	-198.150.11.0	0	E0
C	-198.150.12.0	0	E1
C	-198.150.13.0	0	S0
R	-198.150.14.0	1	S0
R	-198.150.15.0	1	S0

Таблица маршрутизации

Откуда получен	Сетевой адрес	Счетчик узлов	Интерфейс
C	-198.150.21.0	0	E0
C	-198.150.22.0	0	E1
C	-198.150.23.0	0	S1
R	-198.150.24.0	1	S1
R	-198.150.25.0	1	S1

Рис. 10.21. Таблицы маршрутизации

Маршрутизаторы хранят и обновляют следующую важную информацию в таблицах маршрутизации:

- **тип протокола —** информацию о протоколе маршрутизации, создавшем запись в таблице маршрутизации;
- **связка получатель/следующий узел** сообщает маршрутизатору о том, что определенный получатель либо подключен непосредственно, либо может быть достигнут через другой маршрутизатор, называемый *следующим транзитным узлом (next hop)*, находящийся на пути к пункту назначения. Маршрутизатор анализирует адрес получателя во входящих пакетах и сравнивает его на соответствие с записями в таблице маршрутизации;
- **метрики маршрутизации**. Различные протоколы маршрутизации используют разные метрики, которые помогают определить предпочтительность маршрута. Например, протокол RIP использует *счетчик транзитных узлов (hop count)* в качестве метрики маршрутизации. Протокол IGRP использует пропускную способность, загрузку канала, суммарную задержку передачи и надежность для формирования комплексного значения метрики. Более подробно метрики и протоколы маршрутизации обсуждаются во второй части книги "Курс CCNA2: маршрутизаторы и основы маршрутизации";
- **выходной интерфейс —** интерфейс, через который должны быть отправлены данные, чтобы достичь пункта назначения.

Маршрутизаторы взаимодействуют друг с другом посредством передачи сообщений-анонсов для поддержки таблиц маршрутизации. В зависимости от протокола маршрутизации такие обновления маршрутных таблиц могут отправляться либо периодически, либо при изменении топологии сети. Протокол также определяет, нужно ли в анонсе отправить полную таблицу маршрутизации или только информацию об изменившемся маршруте. Используя анонсы, получаемые от соседей, маршрутизатор создает и поддерживает свою таблицу маршрутизации в актуальном состоянии.

Алгоритмы маршрутизации и метрики

Протоколы маршрутизации выбираются, исходя из характеристик, перечисленных ниже.

- **Оптимальность** описывает способности протокола и алгоритма по выбору наиболее оптимального маршрута на основании метрик и их весовых значений, используемых при расчетах. Например, некий протокол может использовать счетчик узлов и задержки для определения метрик; задержки имеют более высокий вес при учете окончательного значения, но зато их сложнее рассчитать.
- **Простота и низкие накладные расходы.** Идеальная эффективность работы алгоритма маршрутизации может быть достигнута, когда загрузка процессора и памяти маршрутизатора минимальны. Эта характеристика важна для масштабируемости сети, которая в предельном случае может быть расширена до размеров сети Internet.

- **Устойчивость и надежность.** Алгоритм маршрутизации должен корректно функционировать даже при наличии нестандартных и непредвиденных обстоятельств, таких, как сбой оборудования, высокая загрузка и ошибки эксплуатации.
- **Быстрая конвергенция.** Конвергенцией называется процесс установления договоренности между всеми маршрутизаторами об имеющихся маршрутах. Когда в сети происходят события, оказывающие влияние на доступность маршрутизатора, для установления повторного соединения требуются перерасчеты. Алгоритмы маршрутизации, не обладающие быстрой конвергенцией, могут вызвать сбой или значительную задержку при доставке информации.
- **Гибкость.** Алгоритм и протокол маршрутизации должны быстро адаптироваться к разнообразным изменениям в сети. Изменениями в сети считаются изменения в состоянии устройств, в частности, маршрутизаторов, изменение пропускной способности каналов, изменение размера очередей или сетевой задержки.
- **Масштабируемость.** Некоторые протоколы разработаны таким образом, что могут быть масштабируемы лучше других. Важно помнить, что если планируется расширение сети (или такая возможность в принципе предусматривается), следует отдать предпочтение протоколу EIGRP, нежели RIP.

Первоочередная задача *алгоритма маршрутизации* при обновлении таблицы маршрутизации состоит в определении наилучшей информации, которая должна быть внесена в таблицу. Алгоритмы маршрутизации используют различные метрики для определения наилучшего маршрута, но каждый алгоритм интерпретирует выбор лучшего варианта пути по-своему. Алгоритм маршрутизации рассчитывает число, называемое метрикой, для каждого сетевого маршрута. Сложные алгоритмы маршрутизации могут основывать выбор маршрута на основе нескольких параметров, объединяя их в одну общую метрику, как показано на рис. 10.22. Чем меньше метрика, тем лучше выбранный маршрут.

Метрики могут быть вычислены на основе одной или нескольких характеристик. Наиболее часто в алгоритмах маршрутизации используются параметры метрики, которые перечислены ниже.

- **Ширина полосы пропускания** представляет собой средство оценки объема информации, который может быть передан по каналу связи (канал Ethernet со скоростью 10 Мбит/с более предпочтителен, чем выделенная линия со скоростью 64 Кбит/с).
- **Задержка** — промежуток времени, необходимый для перемещения пакета по каждому из каналов связи от отправителя получателю. Задержка зависит от пропускной способности промежуточных каналов, размера очередей в портах маршрутизаторов, загрузки сети и физического расстояния.
- **Загрузка** — объем операций, выполняемых сетевым устройством, таким, как маршрутизатор, или средняя загруженность канала связи.

- **Надежность** обычно обозначает относительное значение количества ошибок для каждого из каналов связи.
- **Счетчик транзитных узлов —** количество маршрутизаторов, через которые должен пройти пакет, прежде чем достигнет пункта назначения. Когда пакет проходит через маршрутизатор, значение счетчика узлов увеличивается на единицу. Путь, для которого значение счетчика узлов равно четырем, означает, что данные, отправленные по этому маршруту, пройдут через четыре маршрутизатора, прежде чем будут получены адресатом. Если существует несколько путей, маршрутизатор выбирает тот, для которого значение счетчика узлов наименьшее.
- **Стоимость** — значение, обычно вычисляемое на основе пропускной способности, денежной стоимости или других единиц измерения, назначаемых администратором.

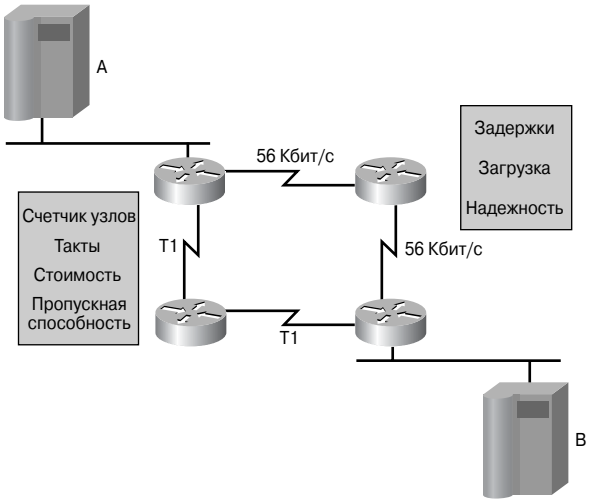


Рис. 10.22. Метрики маршрутизации

Внутренние и внешние протоколы маршрутизации

Маршрутизаторы используют протоколы маршрутизации для обмена маршрутной информацией. Иными словами, протоколы маршрутизации определяют, как маршрутизируются протоколы передачи данных (т.е. маршрутизируемые). Как показано на рис. 10.23, двумя семействами протоколов маршрутизации являются *протоколы внутренних шлюзов (Interior Gateway Protocol — IGP*) и *протоколы внешних шлюзов (Exterior Gateway Protocols — EGP*). Классификация всех протоколов по этим двум семействам основана на принципе их работы по отношению к автономным системам.

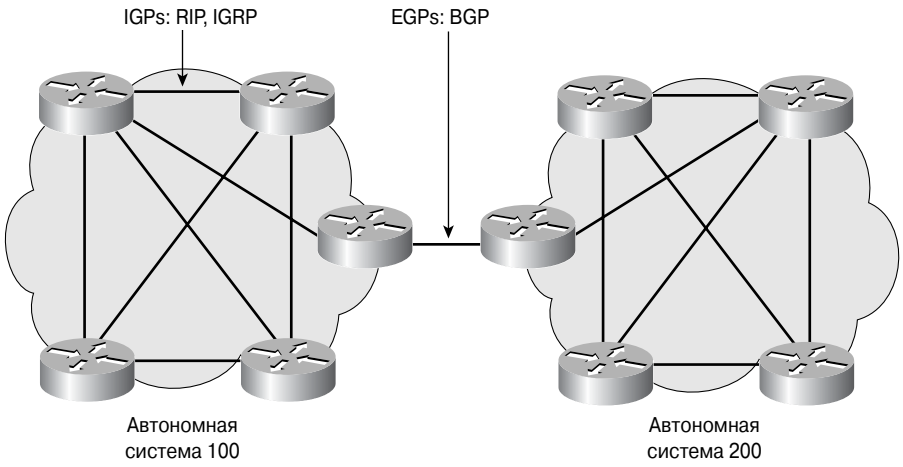


Рис. 10.23. Протоколы EGP и IGP

Автономной системой (Autonomous System — AS) называется сеть или группа сетей, находящихся под единым административным контролем, как, например, домен `Cisco.com`. Автономная система состоит из маршрутизаторов, которые для внешнего мира (т.е. для других сетей) выглядят как единая сеть. Агентство по выделению имен и уникальных параметров протоколов Internet (Internt Assigned Numbers Authority — IANA) выделяет номера автономных систем региональным регистраторам. Таким регистратором для Америки, стран Карибского бассейна и Африки является организация ARIN (American Registry for Internet Numbers — Американский регистратор номеров сети Internet, адрес — `hostmaster@arin.net`), для Европы — RIPE-NCC[2] (Reseaux IP Europeens Network Coordination Centre — сетевой координационный центр RIPE, адрес — `ncc@ripe.net`), для стран Азиатско-тихоокеанского региона — AP-NIC (Asia Pacific Network Information Centre — сетевой информационный центр азиатско-тихоокеанского региона, адрес — `admin@apnic.net`). Такие автономные системы описываются шестнадцатибитовым номером. При настройке таких протоколов маршрутизации, как BGP, требуется указать назначенный уникальный номер автономной системы.

Протоколы класса IGP маршрутизируют данные внутри автономных систем. К классу IGP относятся следующие протоколы маршрутизации:

- протоколы RIP и RIP V2;
- IGRP;
- EIGRP;
- OSPF;

[2] *Домены .ru официально регулируются Российским НИИ Развития Общественных Сетей (Russian Institute for Public Networks — RIPN),* `http://www.ripn.net`. *— Прим. ред.*

- протокол обмена данными между промежуточными системами (Intermediate system-to-Intermediate System — IS-IS).

Протоколы класса EGP маршрутизируют данные между автономными системами. Протокол BGP является наиболее широко известным представителем класса EGP.

Дистанционно-векторные и протоколы маршрутизации с учетом состояния каналов

Протоколы маршрутизации могут подразделяться по самым разным критериям, например, по сфере применения, т.е. по принадлежности к EGP- или IGP-типу. Другой классификацией, описывающей протоколы маршрутизации, может быть деление по используемым алгоритмам: протокол использует дистанционно-векторный (distanse-vector) алгоритм или работает с учетом состояния канала (link-state). Если принадлежность маршрутизаторов к EGP- или IGP-типу описывает их физическое взаимодействие, то использование алгоритмов маршрутизации по вектору расстояния или состоянию канала описывает характер взаимодействия маршрутизаторов между собой при рассылке маршрутных обновлений.

Дистанционно-векторные протоколы

Алгоритм дистанционно-векторной маршрутизации определяет направление (вектор) и расстояние (счетчик узлов) для каждого из каналов связи, образующих сеть. При использовании этого алгоритма маршрутизатор периодически (например, каждые 30 секунд) пересылает всю или часть своей таблицы маршрутизации своим соседям. Периодические обновления рассылаются маршрутизатором, использующим дистанционно-векторный алгоритм, даже если не произошли никакие изменения в сети. Получив таблицу маршрутизации от своего соседа, маршрутизатор может проверить уже известные маршруты и внести необходимые изменения на основе полученного обновления. Такой процесс иногда называют "маршрутизацией по слухам", поскольку представление маршрутизатора о структуре сети базируется на данных его соседей. Дистанционно-векторные протоколы маршрутизации основаны на алгоритме Беллмана-Форда (Bellman-Ford) и используют его для поиска наилучшего маршрута.

Дистанционно-векторный алгоритм служит основой для следующих протоколов (рис. 10.25):

- для *протокола маршрутной информации* (Routing Information Protocol — RIP) — одного из наиболее широко распространенных протоколов IGP-типа, использующего в качестве метрики счетчик узлов;
- для *протокола маршрутизации внутреннего шлюза* (Interior Gateway Routing Protocol — IGRP); корпорация Cisco разработала этот протокол для маршрутизации в больших гетерогенных сетях;

- для *усовершенствованного протокола маршрутизации внутреннего шлюза* (Enhanced Interior Gateway Routing Protocol — EIGRP), представляющего собой улучшенную версию IGRP от корпорации Cisco; этот протокол имеет исключительно быструю конвергенцию, работает значительно более эффективно, чем его предшественник, и сочетает в себе все преимущества дистанционно-векторных алгоритмов и протоколов с учетом состояния каналов.

Протоколы маршрутизации по состоянию каналов

Протоколы маршрутизации, использующие *алгоритм с учетом состояния каналов*, были разработаны для преодоления ограничений, связанных с использованием дистанционно-векторных протоколов. Алгоритм с учетом состояния канала дает возможность протоколам быстро реагировать на изменения сети, рассылать обновления только в случае появления изменений и рассылать периодические обновления (называемые обновлениями состояния канала) через большие промежутки времени, примерно один раз каждые 30 минут.

Когда состояние канала изменяется, устройство, обнаружившее такое изменение, формирует извещение о состоянии канала (Link-State Andvertisement — LSA), относящееся к этому каналу (маршруту), и рассылает его всем соседствующим маршрутизаторам. Каждый маршрутизатор получает копию извещения о состоянии канала и на этом основании обновляет свою базу состояния каналов (топологическую базу), после чего пересылает копию извещения всем своим соседям. Такая массовая рассылка извещения нужна, чтобы гарантировать, что все маршрутизаторы обновят свои базы данных и создадут обновленную таблицу маршрутизации, которая отражает новую топологию (рис. 10.24).

База данных состояния канала используется для обнаружения наилучшего сетевого пути. Маршрутизация с учетом состояния канала основана на алгоритме первоочередного определения кратчайшего маршрута (Shortest Path First — SPF) Дейкстра (Dijkstra) для построения SPF-дерева, на основе которого принимается решение о том, какой маршрут является наилучшим. Наилучший (кратчайший) маршрут выбирается из дерева первоочередного определения кратчайшего маршрута и помещается в таблицу маршрутизации.

Примерами протоколов, использующих алгоритм с учетом состояния каналов, являются OSPF и IS-IS (рис. 10.25).

Интерактивная презентация: дистанционно-векторные протоколы и протоколы маршрутизации по состоянию каналов
Эта презентация позволит закрепить знания о протоколах маршрутизации, в частности, еще раз повторить отличия между двумя классами протоколов.

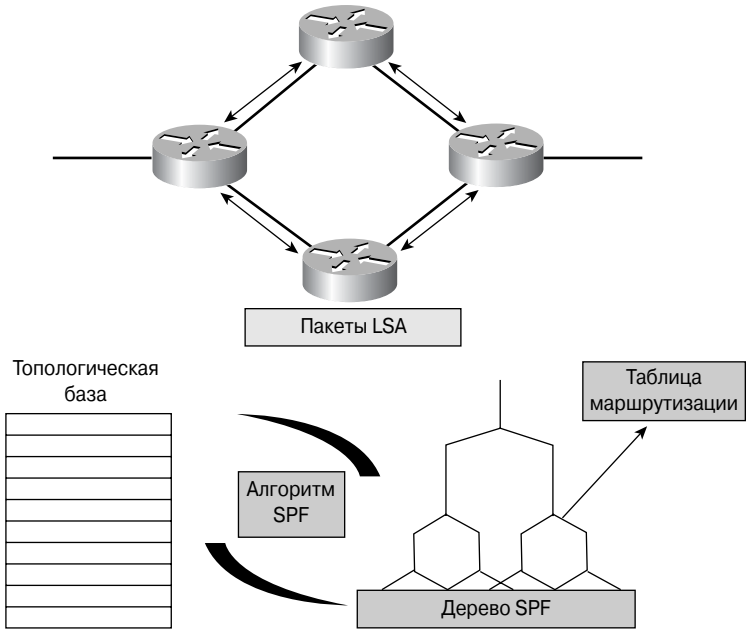


Рис. 10.24. Протоколы маршрутизации на основе состояния канала

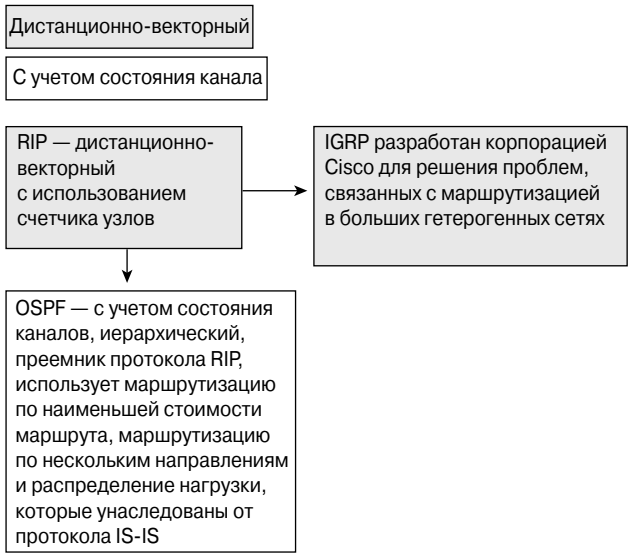


Рис. 10.25. Основные характеристики наиболее распространенных протоколов маршрутизации

Протоколы маршрутизации

В этом разделе описаны метрики, загрузка сети и другие важные характеристики наиболее широко используемых протоколов маршрутизации.

Протокол RIP

Протокол маршрутной информации (Routing Information Protocol — RIP) использует счетчик количества транзитных узлов для определения направления и расстояния для любого из каналов сети (рис. 10.26). Если существуют несколько маршрутов к получателю, протокол RIP выберет тот из них, который имеет наименьшее значение счетчика транзитных узлов. Поскольку счетчик является единственной метрикой, используемой протоколом RIP, выбранный маршрут далеко не всегда оказывается кратчайшим. Протокол RIP версии 1 позволяет использовать только классовую (classfull) маршрутизацию. Это означает, что все сетевые устройства должны иметь одинаковую маску сети, поскольку RIP версии 1 не включает в маршрутные обновления информацию о ней.

Протокол RIP версии 2 использует так называемую *префиксную маршрутизацию (prefix routing*) и пересылает маску сети вместе с анонсами таблиц маршрутизации: именно за счет этой функции обеспечивается поддержка бесклассовой маршрутизации. Благодаря протоколам бесклассовой маршрутизации можно использовать подсети с разной длины масками внутри одной и той же сети. Использование масок подсети разной длины внутри одной сети называется технологией масок переменной длины (Variable-Length Subnet Mask —VLSM).

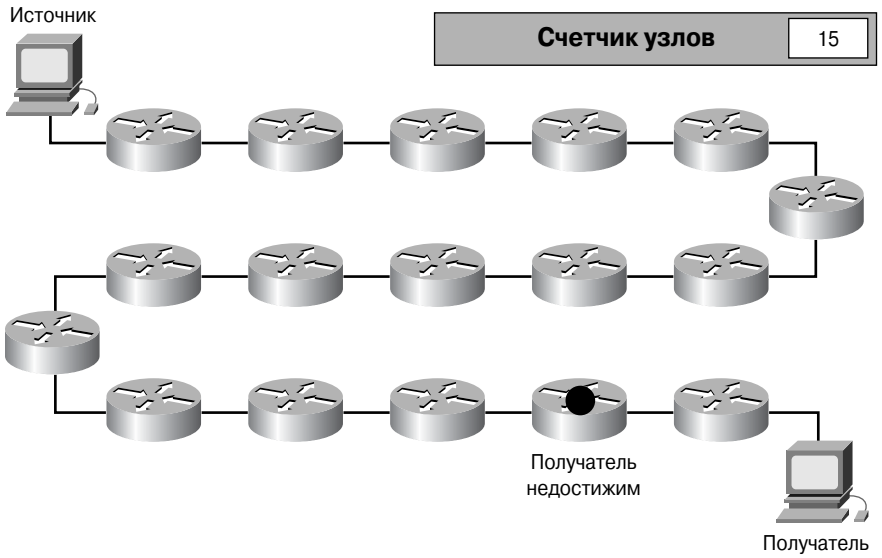


Рис. 10.26. Протокол RIP использует в качестве метрики счетчик транзитных узлов

Протокол IGRP

Протокол маршрутизации внутреннего шлюза (Interior Gateway Routing Protocol — IGRP), разработанный корпорацией Cisco, использует дистанционно-векторный алгоритм и предназначен для решения проблем, возникающих при маршрутизации в больших сетях, где невозможно использовать такие протоколы, как RIP. Протокол IGRP способен выбирать самый быстрый путь на основе задержки, пропускной способности, загрузки и надежности канала. Стандартно протокол IGRP использует в качестве 24-битовых метрик только пропускную способность и задержку. Этот протокол имеет значительно большее максимальное значение счетчика узлов, чем протокол RIP, что дает возможность использовать его в более крупных сетях. Протокол IGRP позволяет использовать только классовую маршрутизацию.

Протокол EIGRP

Так же, как и IGRP, протокол EIGRP (Enhanced Interior Gateway Routing Protocol — расширенный протокол маршрутизации внутреннего шлюза) был разработан корпорацией Cisco и является ее фирменным продуктом. Этот протокол — усовершенствованная версия протокола IGRP, использует 32-битовые метрики. В частности, протокол EIGRP очень эффективен благодаря более быстрой конвергенции и низкому потреблению пропускной способности. Он является усовершенствованным вариантом протокола, работающего на основе дистанционно-векторного алгоритма. Протокол EIGRP также использует некоторые функции алгоритмов с учетом состояния канала. Вот почему использование термина *гибридный* тоже вполне законно при описании протокола IGRP.

Протокол OSPF

Открытый протокол поиска кратчайшего пути (Open Shortest Path First — OSPF) использует алгоритм маршрутизации по состоянию каналов. Проблемная группа проектирования Internet (IETF) разработала OSPF в 1988 году. Самая последняя версия этого протокола, OSPF версии 2, описана в спецификации RFC 2328. OSPF является протоколом IGP-типа, что означает, что он распространяет маршрутную информацию между маршрутизаторами, находящимися в единой автономной системе. Протокол OSPF был разработан для использования в больших сетях, в которых невозможно использование протокола RIP.

Протокол IS-IS

Протокол обмена маршрутной информацией между промежуточными системами (Intermediate System-to-Intermediate System — IS-IS) использует алгоритм маршрутизации по состоянию канала для *стека протоколов* модели OSI. Он распространяет маршрутную информацию для протокола сетевого обслуживания (Connectionless Network Protocol — CLNP), для соответствующих ISO-служб сетевого обслуживания без установления соединения (Connectionless Network Service — CLNS). Интегрированный протокол IS-IS является вариантом реализации протокола IS-IS для маршрутизации нескольких сетевых протоколов. Интегрированный протокол IS-IS объединяет CLNP-маршруты с информацией об IP-сетях и масках подсетей. Благодаря

соединению ISO CLNS и IP-маршрутизации в одном протоколе интегрированный протокол IS-IS предоставляет альтернативу протоколу OSPF при использовании в IP-сетях. Он может быть использован для IP-маршрутизации, ISO-маршрутизации и для комбинации этих двух вариантов.

ВНИМАНИЕ!

Протокол CLNP относится к сетевому уровню эталонной модели OSI и не требует установления виртуального канала перед тем, как будет начата передача данных.

Протокол BGP

Протокол граничного шлюза (Border Gateway Protocol — BGP) является примером протокола EGP-типа. Протокол BGP обеспечивает обмен маршрутной информацией между автономными системами и гарантирует выбор маршрутов без зацикливания. Он является базовым протоколом извещений маршрутизации, использующимся большинством крупных компаний и поставщиками услуг доступа к Internet (ISP). Протокол BGP-4 стал первой версией протокола BGP, в котором встроена *бесклассовая междоменная маршрутизация* (*Classless InterDomain Routing — CIDR*), и первым, использующим механизм агрегации маршрутов. В отличие от распространенных протоколов IGP-типа, таких, как RIP, OSPF и EIGRP, BGP не использует в качестве метрики счетчик узлов, пропускную способность или задержку в сети. Вместо этого протокол BGP принимает решение о выборе маршрута, руководствуясь указанными сетевыми правилами, используя различные маршрутные BGP-атрибуты.

Практическое задание 10.2.9. Покупка небольшого маршрутизатора
Цель этого задания — ознакомиться с существующим разнообразием и ценами на современные сетевые компоненты. В задании делается акцент на использование небольших маршрутизаторов, применяющихся в домашних офисах и для подключения к центральному офису телеработников.

Механизм создания подсетей

В изначальной двухуровневой[3] иерархии сети Internet предполагалось, что каждая подключенная к сети организация будет иметь только одну сеть. Следовательно, каждой организации требовалось бы только одно подключение к сети Internet. Поначалу такое предположение было вполне оправдано и не вызывало опасений. Однако по происшествии некоторого времени компьютерные сети достигли определенного уровня развития и широкого распространения. К 1985 году стало понятно, что предположение о том, что одна организация будет иметь только одну сеть и удовлетворится единственным подключением к глобальной сети Internet, больше не соответствует действительности.

[3] *Т.е. в такой иерархии, когда предполагается, что адрес состоит из двух частей и записывается в виде "сеть.узел". — Прим. ред.*

По мере того как организации начали создавать многочисленные сети, для проблемной группы проектирования Internet (Internet Engineering Task Force — IETF) стало очевидным, что требуется механизм разделения многочисленных логических сетей, появляющихся как подмножества второго уровня сети Internet. В противном случае стало бы невозможным организовать эффективную маршрутизацию данных до определенной конечной системы, принадлежащей организации с многочисленными сетями.

Классы сетевых IP-адресов

Как уже говорилось, сети разных классов могут содержать от 254 до 16,8 млн. адресов узлов. Чтобы наиболее эффективно использовать имеющийся ограниченный запас сетевых IP-адресов, каждая сеть может быть разделена на подсети меньшего размера. На рис. 10.27 показано разделение на сетевую и узловую части адресов сетей разных классов.

Класс A	Сеть	Узел		
Октет	1	2	3	4

Класс B	Сеть		Узел	
Октет	1	2	3	4

Класс C	Сеть			Узел
Октет	1	2	3	4

Класс D	Узел			
Октет	1	2	3	4

Рис. 10.27. Сети классов A-D: сетевая и узловая части

Введение в технологию подсетей и ее обоснование

Чтобы выделить подсеть, биты сетевого узла должны быть переназначены как сетевые биты посредством деления *октета (или октетов)* сетевого узла на части. Такой механизм часто называют *заимствованием битов*, но более точным термином будет *аренда битов*, хотя последний используется очень редко. Процесс деления всегда начинается с крайнего левого бита узла, положение которого зависит от класса IP-адреса.

Помимо повышения управляемости, создание подсетей позволяет сетевым администраторам ограничить широковещательные рассылки и реализовать механизм низкоуровневой безопасности в локальной сети. Безопасность при использовании подсетей в локальных сетях реализуется благодаря тому, что доступ в другие подсети организуется через маршрутизаторы. Маршрутизатор, как рассказывается в главе 22, "Списки управления доступом", может быть настроен таким образом, чтобы разрешить или запретить доступ к подсети на основе различных критериев, реализуя таким образом политику безопасности. Некоторые организации, обладатели сетей классов A и B, обнаружили также, что использование механизма выделения подсетей может принести дополнительные доходы за счет продажи или передачи в аренду ранее не использовавшихся IP-адресов.

На рис. 10.28 показано, как в среде с многочисленными сетями каждая из них подключена к сети Internet посредством единой точки доступа — общего маршрутизатора. Подробности и детали организации внутренней сети несущественны для сети Interenet. С использованием подсетей можно организовать частную сеть, в которой внутренние устройства будут заниматься доставкой данных пользователей. Таким образом, задача устройств сети Internet состоит только в том, как доставить данные сетевому маршрутизатору-шлюзу, посредством которого частная сеть подключена к глобальной. Внутри частной сети узловая часть IP-адреса может быть разделена на части для создания подсетей.

Поскольку *адрес подсети* формируется из узловой части адреса класса A, B или C, он назначается локально, обычно местным сетевым администратором. Кроме того, как и остальные части IP-адреса, каждый адрес подсети должен быть уникальным внутри области их использования (рис. 10.29).

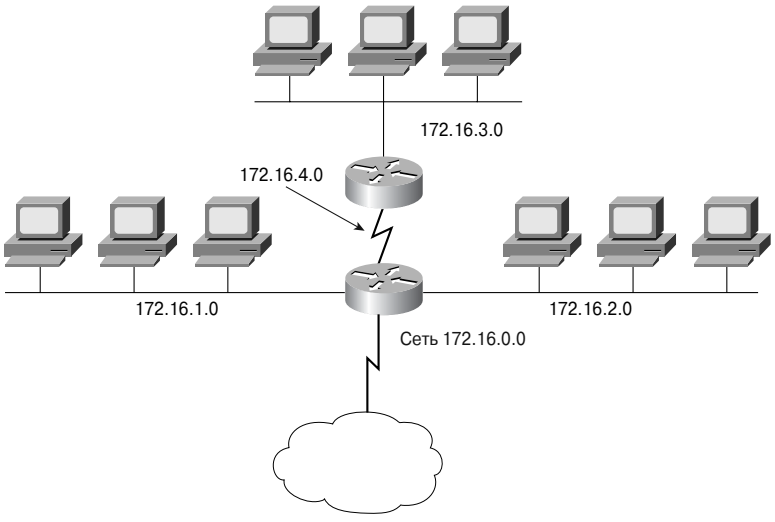


Рис. 10.28. Подсети

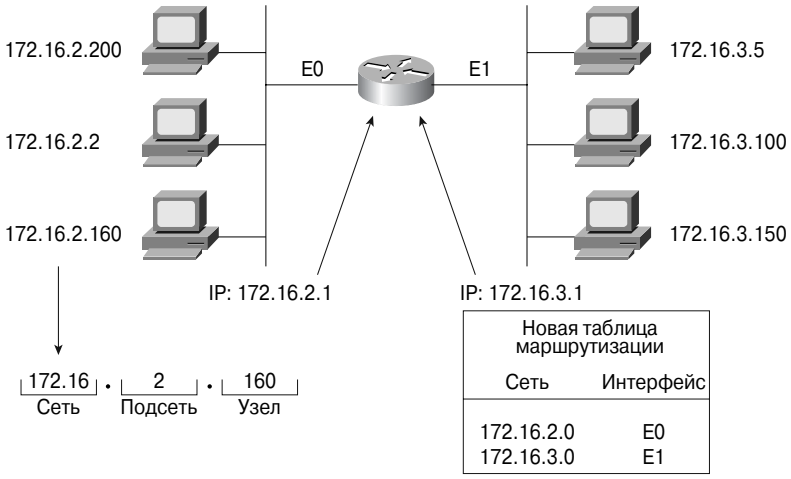


Рис. 10.29. Адреса подсетей

Использование подсетей часто бывает необходимо при объединении локальных сетей с целью создания единой распределенной сети. Например, при объединении двух локальных сетей, расположенных в географически удаленных точках, можно назначить уникальные подсети каждой из локальных сетей и каналу распределенной сети между ними. В таком случае могут быть использованы два маршрутизатора (по одному в каждой из сетей) для маршрутизации пакетов между локальными сетями (подсетями).

Другой важной причиной использования подсетей является необходимость в уменьшении размеров широковещательных доменов. Широковещательные пакеты рассылаются всем узлам в сети или подсети. Когда широковещательный трафик начинает расходовать значительную часть доступной полосы пропускания, сетевой администратор может принять решение об уменьшении размеров широковещательного домена.

Внешний мир "видит" локальную сеть как единую сеть, ничего не зная о ее внутренней структуре. Такой подход позволяет уменьшить таблицы маршрутизации и эффективно их использовать. Получив локальный адрес узла 192.168.10.14, внешний мир за пределами локальной сети использует только объявленный основной сетевой адрес 192.168.10.0. Причина этого в том, что локальный адрес 192.168.10.14 действителен только в пределах локальной сети 192.168.10.0. В других местах он работать не будет.

Адрес подсети включает сетевую часть адреса классов A, B и C плюс поле подсети и поле узла. Эти поля создаются на основе оригинального IP-адреса заимствованием битов из узловой части адреса и присоединением к исходной сетевой части адреса. Как показано на рис. 10.30-10.32, возможность деления оригинальной узловой части адреса на новые подсети и адреса узлов предоставляет гибкость в выборе схемы адресации для сетевых администраторов. Это означает, что у сетевого администратора

есть более широкий выбор при выборе схемы адресации как изначально, так и при расширении сети.

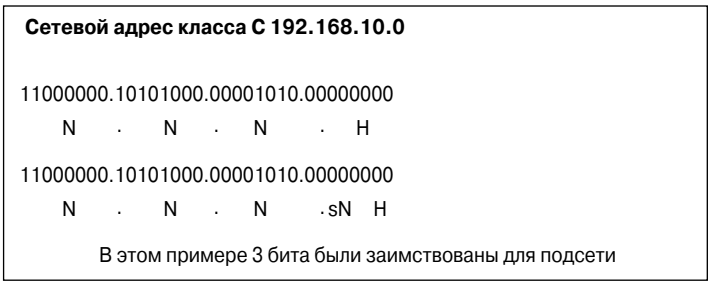


Рис. 10.30. Деление узлового октета адреса класса C

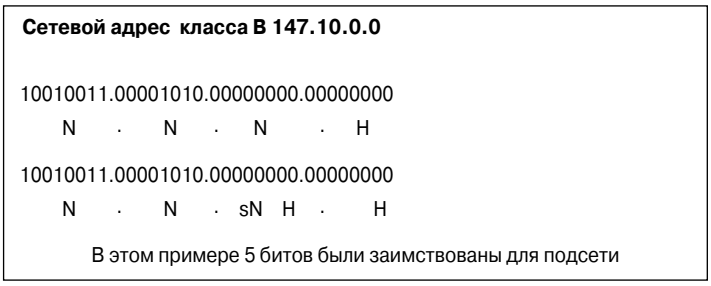


Рис. 10.31. Деление узлового октета адреса класса B

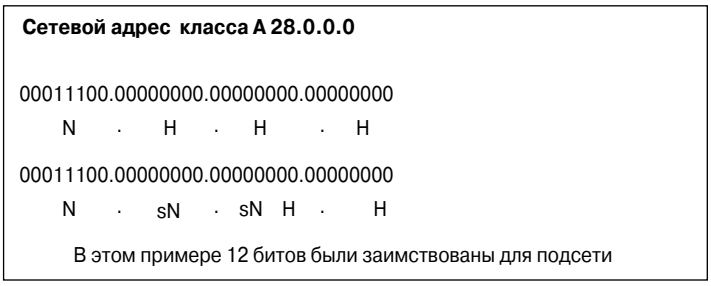


Рис. 10.32. Деление узлового октета адреса класса A

Назначение маски подсети

Выбор необходимого количества битов для создания подсети зависит от требуемого максимального количества узлов в подсети. Чтобы вычислить результат заимствования определенного количества узловых битов для создания подсети, необходимо иметь базовые знания из области двоичной математики и помнить битовые значения в каждой из позиций октета, как показано в табл. 10.2.

Таблица 10.2. Расчет подсети: позиция бита и соответствующее ему десятичное значение

Бит	1	2	3	4	5	6	7	8
Значение	128	64	32	16	8	4	2	1

Независимо от класса IP-адреса, последние 2 бита в последнем октете никогда не могут быть использованы для формирования подсети. Они называются *наименее значимыми битами*. Заимствование всех доступных битов, за исключением двух последних, позволяет создать подсеть, которая содержит только два узла. Такой способ используется на практике для экономии адресов при адресации последовательных связей между маршрутизаторами. Однако для работающих локальных сетей это вызвало бы недопустимые расходы на оборудование.

Чтобы создать *маску подсети*, дающую маршрутизатору информацию, необходимую для вычисления адреса подсети, которой принадлежит конкретный узел, необходимо выбрать столбец из таблицы с нужным количеством битов и в качестве значения маски воспользоваться числом строкой выше из того же столбца, как показано в табл. 10.3. Это значение получено в результате сложения двоичных значений для знакомест используемых битов. Как показано в табл. 10.3, если заимствованы 3 бита, маска подсети для сети класса C будет равна 255.255.255.224. При использовании формата записи маски с обратной косой чертой он может быть представлен как "/27". Число, указанное после символа обратной косой черты, представляет собой количество битов, составляющих адрес сети, плюс биты, использующиеся для маски подсети.

Таблица 10.3. Расчет подсети: два формата маски подсети

Формат с обратной косой чертой	/25	/26	/27	/28	/29	/30	—	—
Маска	128	192	224	240	248	252	254	255
Бит	1	2	3	4	5	6	7	8
Значение	128	64	32	16	8	4	2	1

Чтобы определить требуемое количество битов, разработчик сети должен рассчитать, какое максимальное число узлов будет в подсети, и общее количество подсетей. В качестве примера предположим, что необходимо разместить по 30 узлов в 5-ти подсетях. Чтобы определить необходимое количество битов для переназначения, воспользуемся строкой "Количество используемых узлов" табл. 10.4. Так, для использования 30-ти узлов требуются 3 бита. Таким образом будет создано 6 подсетей, что также удовлетворяет указанным выше требованиям. Следует помнить, что разница в количестве доступных узлов и полном количестве возникает из-за того, что первый доступный адрес является идентификатором сети, а последний — ее широковещательным адресом. Классовая маршрутизация не предоставляет механизм

использования соответствующих подсетей, в то время как при бесклассовой маршрутизации множество таких "потерянных" адресов доступно для использования, как показано в табл. 10.4. Глядя на таблицу, можно также оценить, какое количество подсетей и узлов будет потеряно, если бесклассовая маршрутизация не используется.

Таблица 10.4. Расчет подсети: подсети и узлы

Формат с обратной косой чертой	/25	/26	/27	/28	/29	/30	—	—
Маска	128	192	224	240	248	252	254	255
Бит	1	2	3	4	5	6	7	8
Значение	128	64	32	16	8	4	2	1
Всего подсетей		4	8	16	32	64		
Доступные подсети		2	6	14	30	62		
Всего узлов		64	32	16	8	4		
Количество используемых узлов		62	30	14	6	2		

Еще одним способом вычислить маску подсети и количество доступных подсетей и узлов является использование формул, которые приведены и объяснены ниже.

Количество доступных подсетей равно 2 в степени, равной количеству используемых для формирования подсети битов, минус 2:

$(2^{\text{количество заимствованных битов}}) - 2 =$ количество используемых подсетей.

Например, при заимствовании трех битов из узловой части сети класса C $2^3 - 2 = 6$ — количество используемых подсетей.

Количество доступных узлов равно 2 в степени, равной количеству оставшихся от заимствования битов, минус 2:

$(2^{\text{оставшиеся биты}}) - 2 =$ количество используемых узлов.

Например, при заимствовании трех битов из узловой части сети класса C для адресации узлов будут использоваться 5 битов, следовательно, количество узлов в каждой подсети равно $2^5 - 2 = 30$.

Создание подсети

Для создания подсети необходимо расширить часть адреса, с которой оперируют маршрутизаторы. В сети Internet устройства оперируют с сетью как с единым целым, согласно классам адресов A, B или C, которые задаются восемью, шестнадцатью или

двадцатью четырьмя битами в маске (т.е. номером сети). Поле подсети описывает дополнительные биты, давая возможность локальным маршрутизаторам оперировать разными подсетями внутри единой, большой сети.

В маске подсети используется тот же формат, что и в IP-адресе. Иными словами, маска подсети состоит из четырех октетов, а длина ее составляет 32 бита. Сетевая часть маски подсети, как и часть, определяющая подсеть, состоит из всех единиц, а узловая ее часть заполнена нулем. Стандартно, если ни один бит не заимствован для разбиения сети на подсети, маска для сети класса B выглядит как 255.255.0.0. Если заимствованы 8 битов, соответствующая маска будет иметь вид 255.255.255.0, как показано на рис. 10.33 и 10.34. Поскольку в адресе класса B выделены два октета под адреса узлов, для задания маски подсети может быть заимствовано не более 14 битов. В сети класса C используются только 8 битов для поля узла. Следовательно, для задания маски подсети может быть заимствовано не более 6 битов.

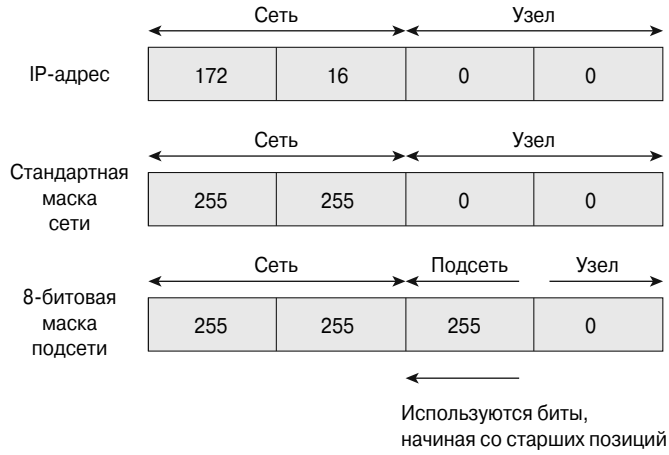


Рис. 10.33. Адреса сети и узла

Поле подсети всегда следует непосредственно за номером сети. Такое требование означает, что заимствовать можно первые *n* битов из стандартного поля узлов, где *n* — необходимая длина поля создаваемой подсети, как показано на рис. 10.35. Маска подсети является инструментом, который помогает маршрутизатору в определении сетевой (и используемой маршрутизатором) части адреса и его узловой части.

128	64	32	16	8	4	2	1		
1	0	0	0	0	0	0	0	=	128
1	1	0	0	0	0	0	0	=	192
1	1	1	0	0	0	0	0	=	224
1	1	1	1	0	0	0	0	=	240
1	1	1	1	1	0	0	0	=	248
1	1	1	1	1	1	0	0	=	252
1	1	1	1	1	1	1	0	=	254
1	1	1	1	1	1	1	1	=	255

Рис. 10.34. Схема двоичных преобразований

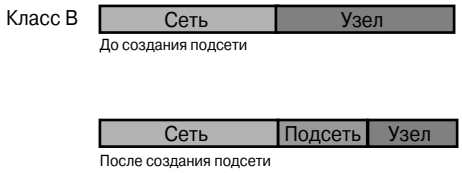


Рис. 10.35. Создание подсети в адресе класса B

Дополнительная информация: определение размера маски подсети

Как уже говорилось, в маске подсети все биты в сетевой части (их количество определяется классом сети) и в части, которая описывает подсеть, равны 1, а все оставшиеся биты маски равны 0, поскольку они относятся к узловой части адреса.

Стандартно, если нет заимствования битов, маска подсети для сети класса B имеет значение 255.255.0.0; такая запись эквивалентна тому, что в первых 16 битах адреса, описывающих номер сети класса B, установлены единицы и нули в оставшихся 16 битах.

Если для задания поля подсети заимствованы 8 битов, маска подсети будет содержать дополнительные единицы еще в 8 битах и станет равна 255.255.255.0. Например, если маска подсети используется с адресом 130.5.2.144 для сети класса B (8 битов заимствованы для подсети), маршрутизатор будет знать, что такие пакеты следует направлять сети 130.5.2.0, а не сети с адресом 130.5.0.0, как это показано на рис. 10.36.

Рассмотрим другой пример: сеть класса C, адрес узла равен 197.15.22.131 и маска равна 255.255.255.224. Использование в последнем октете маски числа 224 (11100000 в двоичном виде) означает, что 24-битовый адрес сети класса C расширен на 3 бита, что в сумме дает 27 битов. Число 131 в последнем октете описывает третий, доступный для использования адрес узла в сети с адресом 197.15.22.128, как показано на рис. 10.37. Маршрутизаторы в сети Internet (которые не знают о маске подсети) отвечают только за доставку пакетов в сеть 197.15.22.0. Маршрутизаторы внутри этой сети, знающие о маске подсети, принимают решение об окончательной маршрутизации, используя 27 битов маски для вычисления адреса сети.

Рис. 10.36. Использование маски подсети для адреса класса B

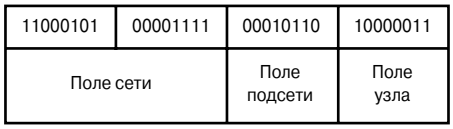


Рис. 10.37. Использование маски подсети: адрес класса C

Расчет маски подсети и IP-адреса

В процессе заимствования битов из поля узла важно уметь подсчитать количество дополнительных подсетей, создаваемых каждый раз при заимствовании каждого дополнительного бита. Мы уже говорили, что заимствование одного бита невозможно; наименьшее допустимое значение равно двум. Заимствуя два бита, можно создать четыре доступные подсети (2×2) (однако при этом следует помнить, что есть еще две зарезервированные не используемые подсети). При заимствовании каждого следующего бита из поля узла количество доступных подсетей увеличивается в 2 раза. Восемь подсетей создаются при заимствовании трех битов (2×2×2). Шестнадцать подсетей появятся в результате заимствования 4-х битов (2×2×2×2). Из перечисленных примеров, а также из схемы двоичных преобразований, показанной на рис. 10.34, можно сделать вывод, что каждый раз при заимствовании дополнительного бита количество доступных подсетей удваивается.

Расчет количества узлов в подсети

Каждый раз при заимствовании одного бита из поля узла количество битов, которые используются для указания номеров узлов, уменьшается. Строго говоря, каждый раз при заимствовании нового бита из поля узла количество адресов узлов, которые могут быть назначены, уменьшается вдвое.

Чтобы понять, как это происходит, рассмотрим для примера сетевой адрес класса C. Без маски подсети все 8 битов последнего октета используются в поле узла. Следовательно, могут быть использованы 256 (2^8) адресов для назначения узлам (254 за вычетом двух, которые, как известно, не могут быть использованы). Предположим теперь, что данная сеть класса C разделена на подсети. В случае, если заимствованы два бита из стандартных восьми, поле узла уменьшится до шести битов. Если воспользоваться всеми возможными комбинациями нулей и единиц в оставшихся шести битах, получится, что полное число доступных узлов, которые могут быть назначены узлам в каждой из подсетей, уменьшится до 64-х (2^6). Количество адресов узлов, которые могут быть использованы, равно 62.

Если в примере с адресом сети класса C заимствуются 3 бита, количество доступных битов в поле узла сократится до 5-ти и общее количество адресов узлов, которые могут быть назначены в каждой из подсетей, уменьшится до 32-х (2^5). Количество адресов узлов, которые могут быть использованы, равно 30-ти.

Количество возможных адресов узлов связано с количеством создаваемых подсетей. Например, для сети класса C и маски подсети 255.255.255.224 3 бита (224 в десятичной форме, что соответствует 11100000 в двоичной) заимствованы из поля узла. Подобным образом могут быть созданы 6 подсетей (8 - 2), в каждой из которых можно использовать 30 (32 - 2) адресов узлов.

Разбиение на подсети сетей класса A и B

Процесс разбиения на подсети сетей класса A и B полностью аналогичен процедуре, выполняемой для сетей класса C, но все же он немного сложнее, поскольку используется большее количество битов. Для использования в подсетях в сети класса A доступны 22 бита, в сети класса B — 24 бита, как показано на рис. 10.4139 и 10.3941.

Заимствование 12-ти битов из узловой части адреса сети класса B создает сетевую маску 255.255.255.240, или в другом обозначении — префикс /28. Все восемь битов третьего октета были использованы для создания маски, поэтому его значение равно 255-ти — максимальное значение восьми единичных битов. В четвертом октете были использованы только четыре бита, следовательно, его значение будет равно 240. Следует помнить, что маска подсети представляет собой сумму заимствованных битов и фиксированных битов сетевой части адреса.

Заимствование 20-ти битов в адресе класса A для создания подсети создает сетевую маску 255.255.255.240, или в другом обозначении — префикс /28. Все восемь битов второго и третьего октетов, а также 4 бита последнего октета в данном случае будут равны 1 и будут принадлежать маске подсети.

В рассмотренной ситуации на первый взгляд может показаться, что маски для сетей класса A и B будут абсолютно идентичными. Тем не менее, не зная, для какой сети или, точнее, для сети какого класса рассчитана маска, невозможно сказать, сколько в действительности битов было заимствовано для создания подсети.

Независимо от того, для сети какого класса необходимо рассчитать подсеть, правила расчета будут одинаковы:

общее количество подсетей = $2^{\text{количество заимствованных битов}}$;

общее количество узлов в подсети = $2^{\text{количество оставшихся от заимствования битов}}$;

общее количество используемых подсетей = $2^{\text{количество заимствованных битов}} - 2$;

общее количество используемых узлов в подсети = $2^{\text{количество оставшихся от заимствования битов}} - 2$.

Практическое задание 10.3.5a. Базовые принципы создания подсетей

В этом задании представлен краткий обзор механизма создания подсетей и использующейся в сетях операции логического умножения. По заданному адресу сети и с учетом дополнительных требований необходимо вычислить подходящую маску подсети, общее и доступное для использования количество подсетей и узлов в каждой из них. Кроме того, используя процедуру логического умножения, необходимо определить, является ли адрес получателя локальным или удаленным. И в конце на основании номера сети и маски подсети требуется определить, является ли действительным определенный IP-адрес узла.

Адрес сети класса B 147.10.0.0 (доступно 14 битов)
11001011.00001010.00000000.00000000 N . N . H . H
10010011.00001010.00000000.00000000 N . N . sN . sN H В данном примере 12 битов было заимствовано для создания подсети

Рис. 10.38. Деление сети класса B на подсети

Адрес сети класса A 28.0.0.0 (доступно 22 бита)
00011100.00000000.00000000.00000000 N . H . H . H
00011100.00000000.00000000.00000000 N . sN . sN . sN H В данном примере 20 битов было заимствовано для создания подсети

Рис. 10.39. Деление сети класса A на подсети

Практическое задание 10.3.5b. Создание подсетей для сети класса A
В этом упражнении необходимо проанализировать сетевой адрес класса A для определения количества сетевых битов, выделяемых для создания маски подсети, количества подсетей, узлов в каждой подсети и информации об определенной подсети.

Практическое задание 10.3.5c. Создание подсетей для сети класса B
В этом упражнении необходимо проанализировать сетевой адрес класса B для определения количества сетевых битов, используемых для создания маски подсети, количества подсетей, узлов в каждой подсети и информации об определенной подсети.

Практическое задание 10.3.5d. Создание подсетей для сети класса C
В этом упражнении необходимо проанализировать сетевой адрес класса C для определения количества сетевых битов, которые могут быть использованы для создания маски подсети, количества подсетей, узлов в каждой подсети и информации об определенной подсети.

Вычисление адреса подсети посредством логической операции AND

Как уже говорилось, адрес сети или подсети содержит все нули в поле адреса узла. Для маршрутизации пакета маршрутизатор в первую очередь должен определить адрес сети или подсети получателя. Для этого маршрутизатор выполняет операцию логического умножения (операция AND или логическое “И”) с использованием IP-адреса узла получателя и соответствующей ему маски подсети.

Предположим, что для адресации используется сеть класса B с адресом 172.16.0.0. После оценки потребностей организации было заимствовано 8 битов для создания подсетей. Как было показано ранее, при заимствовании 8 битов маска подсети для сети класса B будет равна 255.255.255.0 (рис. 10.40).

	Сеть	Подсеть	Узел
IP-адрес узла 172.16.2.120	10101100 00010000	00000010	01111000
Маска подсети 255.255.255.0 или /24	11111111 11111111	11111111	00000000
Подсеть	10101100 00010000 172 16	00000010 2	00000000 0

Рис. 10.40. Использование 8-ми битов для задания подсети

Некто, находящийся вне данной сети, посылает пакет получателю с IP-адресом 172.16.2.120. Для определения направления, в котором следует отправить этот пакет, маршрутизатор производит логическое умножение (операция “И”) адреса с маской подсети.

В результате логического умножения двух чисел узловая часть адреса всегда получается равной нулю, и маршрутизатор вычисляет сетевой адрес, включающий подсеть. Таким образом, данные будут отправлены в подсеть с адресом 172.16.2.0, и только последний маршрутизатор, который рассчитывает маршрут, будет знать, что пакет необходимо доставить узлу с номером 120 в данной подсети.

Теперь предположим, что существует сеть с тем же адресом 172.16.0.0. Однако в этот раз заимствуются 7 битов для поля подсети. В двоичном виде маска выглядит для этого случая как 11111111.1111111.11111110.0000000. Как будет выглядеть данное значение в точечно-десятичном формате?

Как и в предыдущем примере, некто посылает пакет, адресованный узлу 172.16.2.120. Чтобы определить, куда следует отправить данные, маршрутизатор снова производит логическое умножение этого адреса и маски подсети. Как и ранее, при логическом умножении двух чисел узловая часть адреса будет равно нулю. В чем же разница между двумя приведенными выше примерами? Все выглядит идентично, по крайней мере, в десятичном виде. Разница состоит в количестве доступных подсетей и узлов в каждой из них. Отличие можно увидеть, только сравнив две разные маски подсети, как это показано на рис. 10.41.

При использовании семи битов в поле подсети можно выделить только 126 подсетей. Сколько узлов будет в таком случае доступно в каждой из подсетей? При 9 битах, используемых для узловой части, могут существовать до 510 узлов в каждой из этих 126-ти подсетей.

	Сеть	Подсеть	Узел
IP-адрес узла 172.16.2.120	10101100 00010000	00000010	01111000
Маска подсети 255.255.254.0 или /23	11111111 11111111	11111110	00000000
Подсеть	10101100 00010000 172 16	00000010 2	00000000 0

Рис. 10.41. Номер сети, расширенный дополнительными семью битами

Презентация: логическая операция “И”
В этой видеопрезентации проиллюстрирована логическая операция “И” (AND), которую выполняют маршрутизаторы над адресами и сетевыми масками.

Презентация: создание подсетей в сети класса C, часть 1
В этой видеопрезентации показан пример разбиения сети класса C на подсети.

Презентация: создание подсетей в сети класса C, часть 2
В этой видеопрезентации показан второй пример разбиения сети класса C на подсети.

Презентация: создание подсетей в сети класса C, часть 3
В этой видеопрезентации показан третий пример разбиения сети класса C на подсети.

Презентация: создание подсетей в сети класса B, часть 1
В этой видеопрезентации показан пример разбиения сети класса B на подсети.

Презентация: создание подсетей в сети класса B, часть 2
В этой видеопрезентации показан второй пример разбиения сети класса B на подсети.

Резюме

В главе была изложена информация по следующим ключевым вопросам:

- IP является протоколом без установления соединения, он не создает выделенный виртуальный канал между отправителем и получателем, перед тем как начать передачу информации;
- протокол IP также является ненадежным, поскольку в нем не содержатся механизмы, которые проверяют, достигли ли данные пункта назначения. Если требуется выполнить соответствующую проверку, то необходимо, чтобы протокол IP работал в связке с каким-либо транспортным протоколом с установлением соединения, например, протоколом TCP. Если же финальная проверка

и безошибочная доставка не требуются, протокол IP может быть использован в комбинации с каким-либо протоколом без установления соединения, например, протоколом UDP;

- службы без установления соединения зачастую называют процессами коммутации пакетов. Службы с установлением соединения зачастую называют процессами коммутации каналов;
- протоколы всех уровней эталонной модели взаимодействия открытых систем (OSI) добавляют контрольную и управляющую информацию в передаваемые данные по мере их продвижения по сети. Такая информация добавляется как в начало, так и в конец блока данных; сам процесс называется инкапсуляцией данных (т.е. упаковкой данных в информацию соответствующего уровня). На третьем уровне модели OSI добавляется сетевая, или логическая, адресная информация; на втором уровне модели добавляется локальная, или физическая, адресная информация;
- маршрутизация третьего уровня и коммутация второго представляют собой основные механизмы пересылки и доставки данных по сети. Изначально маршрутизатор принимает фрейм второго уровня, в котором инкапсулирован пакет третьего уровня, и обрабатывает его. Маршрутизатор должен отбросить информацию фрейма второго уровня, проверить и обработать пакет третьего уровня. Если пакет может быть доставлен локально, маршрутизатор должен инкапсулировать его в новый фрейм, который содержит правильный MAC-адрес в качестве идентификатора получателя. Если же данные должны быть доставлены в другой (удаленный) широковещательный домен, маршрутизатор должен инкапсулировать пакет третьего уровня в новый фрейм второго уровня, который в качестве адреса получателя содержит MAC-адрес следующего по маршруту межсетевого устройства. Таков процесс доставки данных по сети, от одного широковещательного домена другому; в итоге информация будет доставлена нужному конечному узлу;
- маршрутизируемые протоколы, например, IP, используются для транспортировки данных по сети. Протоколы маршрутизации позволяют маршрутизирующим устройствам выбрать оптимальный маршрут пересылки данных от отправителя получателю. Такие маршруты могут быть как статическими, т.е. такими, которые администратор сети вводит в конфигурацию устройства вручную, так и динамическими, т.е. такими, которые маршрутизатор получает посредством протоколов маршрутизации;
- если устройство использует динамический протокол или протоколы маршрутизации, оно обменивается информацией с другими маршрутизаторами посредством анонсов маршрутизации и таким образом поддерживает свои таблицы маршрутов в актуальном состоянии;
- алгоритмы маршрутизации используют метрики для обработки анонсов маршрутов и заполнения таблиц маршрутизации оптимальными (так называемыми наилучшими) маршрутами;

- время конвергенции протокола маршрутизации (или сети, если используются несколько протоколов) — это интервал после изменения в сети, по истечению которого все маршрутизаторы в сети обладают одинаковой информацией о ее структуре и маршрутах;
- протоколы внутреннего шлюза (IGP) используются внутри автономной системы (AS), протоколы же внешнего шлюза (EGP) предназначены для поиска оптимальных маршрутов между системами AS;
- протоколы IGP далее могут быть классифицированы по принципу работы: дистанционно-векторные и с учетом состояния каналов. В классических дистанционно-векторных протоколах маршрутизации периодически рассылаются анонсы маршрутизации, в которых содержится частичная либо полная таблица маршрутизации. В протоколах маршрутизации по состоянию каналов используется механизм LSA в качестве средства передачи анонсов, обновления информации о маршрутизации рассылаются только после изменения топологии сети, а не периодически; полная таблица маршрутизации рассылается значительно реже, чем в дистанционно-векторных протоколах;
- чтобы пакет мог быть передан по сети, устройствам необходимо наличие некоторого механизма, который позволит отличить часть IP-адреса от узловой. Маска адреса, 32-битовая величина, которую часто также называют маской подсети, указывает устройствам, какую часть IP-адреса следует трактовать как сетевую. Стандартная маска для сети класса A равна 255.0.0.0, для сети класса B — 255.255.0.0, а стандартная маска для сети класса C равна 255.255.255.0. С помощью маски подсети существующую стандартную классовую сеть можно разделить на подсети;
- чтобы предоставить сетевым администраторам дополнительную гибкость, сети, в особенности крупные, часто делятся на более мелкие, называемые подсетями. Механизм создания подсетей позволяет сетевым администраторам преодолеть ограничения, связанные с доступностью IP-адресов, с помощью деления целого сетевого адреса на множество подсетей, видимых только в пределах единой сети. Подсети позволяют уменьшить размеры широковещательных доменов, обеспечивают взаимодействие территориально удаленных сегментов локальных сетей посредством маршрутизаторов и повышение уровня безопасности за счет разделения участков локальных сетей;
- созданная администратором маска подсети использует больше битов, чем отведено для оригинальной классовой маски сети, поскольку биты заимствуются из узловой части адреса. Маска подсети состоит из трех частей:
  - оригинального номера сети;
  - адреса подсети, который создается за счет заимствования битов;
  - адреса узла, который формируется оставшимися незаимствованными битами;

- маршрутизаторы используют сетевые маски, чтобы определить адрес сети для входящего пакета. Это достигается с помощью логической операции "И" (AND);
- межсетевые функции сетевого уровня эталонной модели OSI включают в себя сетевую адресацию и выбор наилучшего пути для потока данных.

Обратите внимание на относящиеся к этой главе интерактивные материалы, которые находятся на компакт-диске, предоставленном вместе с книгой: электронные лабораторные работы (e-Lab), видеоролики, мультимедийные интерактивные презентации (PhotoZoom). Эти вспомогательные материалы помогут закрепить основные понятия и термины, изложенные в настоящей главе.

Ключевые термины

IP-адрес — это 32-битовый адрес, назначаемый узлу при использовании протокола TCP/IP. IP-адрес принадлежит одному из пяти классов (A, B, C, D или E) и записывается в виде четырех октетов, разделенных точками (такой формат называется точечно-десятичным). Каждый адрес состоит из номера сети, необязательного номера подсети и номера узла. Адреса сети и подсети совместно используются для маршрутизации, а адрес узла необходим для доставки информации определенному сетевому узлу внутри сети или подсети. Маска подсети используется для извлечения из IP-адреса информации о сети и подсети. Механизм бесклассовой междоменной маршрутизации (Classless InterDomain Routing — CIDR) предоставляет новый способ представления IP-адресов и маски подсети. Этот тип адреса часто называют *Internet-адресом*.

MAC-адрес — это стандартизованный адрес канального уровня, необходимый каждому устройству, подключенному к локальной сети. Все устройства используют MAC-адреса, чтобы найти определенные устройства в сети, а также для создания и обновления таблиц коммутации и структур данных. Длина MAC-адресов составляет 6 байтов, контролируются они Институтом инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers — IEEE). Этот тип адреса также называют *аппаратным адресом* (hardware address), *адресом MAC-уровня* (MAC-layer address) и *физическим адресом* (physical address).

NetBEUI (расширенный пользовательский интерфейс NetBIOS — NetBIOS Extended User Interface) — это усовершенствованная версия протокола NetBIOS, используемого такими операционными системами, как LAN Manager, LAN Server, Windows for Workgroups и Windows NT. NetBEUI формализует транспортные фреймы и добавляет дополнительные функции. Механизм NetBEUI реализует протокол LLC2 модели OSI.

Автономная система — это отдельная сеть или набор сетей, находящихся под единым административным контролем, как, например, домен `Cisco.com`.

Адрес подсети — это часть IP-адреса, задающая подсеть с помощью маски подсети.

Алгоритм представляет собой четко заданные правила или процесс решения определенной проблемы. В области сетевых технологий алгоритмы в основном используются для определения наилучшего маршрута потока данных от конкретного отправителя к заданному получателю.

Бесклассовая междоменная маршрутизация (Classless InterDomain Routing — CIDR) — технология, поддерживаемая протоколом BGP (и многими другими) и основанная на агрегации маршрутов. Маршрутизация CIDR позволяет маршрутизаторам группировать маршруты, сокращая таким образом объем маршрутной информации, хранящейся в базовых маршрутизаторах. Благодаря использованию механизма CIDR несколько сетей могут быть сгруппированы и выступают в виде одного более крупного блока, который выглядит как единое целое для остальных сетей.

Дейтаграмма — логично связанный блок информации, передаваемой в сетевой среде в качестве модуля передачи сетевого уровня без предварительной установки виртуального соединения. IP-дейтаграммы являются основной единицей информации в сети Internet. Термины *ячейка*, *фрейм*, *сообщение*, *пакет* и *сегмент* также описывают способы логической группировки информации на разных уровнях модели OSI и разных технологических циклах.

Дистанционно-векторная маршрутизация представляет собой класс алгоритмов маршрутизации с последовательным подсчетом транзитных переходов пакета между маршрутизаторами на пути следования для расчета связующего дерева кратчайшего пути. Механизм обновления таблиц маршрутизации "дистанционно-векторный алгоритм" требует от каждого маршрутизатора из числа своих непосредственных соседей выслать свои полные таблицы маршрутизации. При использовании данного алгоритма маршрутизации возможно возникновение кольцевых маршрутов, однако механизм расчета маршрутов проще, чем у алгоритмов маршрутизации по состоянию канала. Этот тип маршрутизации основан на алгоритме Беллмана-Форда (Bellman-Ford).

Домен коллизий — область сети Ethernet, внутри которой распространяются сталкивающиеся фреймы. Концентраторы и повторители пропускают коллизии, коммутаторы локальных сетей, мосты и маршрутизаторы — нет.

Маршрутизатор — устройство сетевого уровня, использующее одну или несколько метрик для определения оптимального пути, по которому следует передавать поток данных. Маршрутизаторы передают пакеты между сетями на основе информации сетевого уровня, содержащейся в маршрутных обновлениях. Иногда такие устройства также называются *шлюзами (gateway)*, однако подобное определение шлюза на сегодняшний день является устаревшим.

Маршрутизируемый протокол — любой сетевой протокол, предоставляющий достаточно информации в адресе сетевого уровня, необходимой для передачи пакета от одного узла другому на основе принятой схемы маршрутизации.

Маска подсети — 32-битовые маски в протоколе IP, служат для указания битов IP-адреса, использующихся в адресе подсети. Иногда их называют просто *маской*.

Метрика маршрутизации представляет собой метод, с помощью которого алгоритм маршрутизации определяет, какой из маршрутов предпочтительнее. Информация о метрике хранится в таблицах маршрутизации и передается вместе с маршрутными обновлениями. В качестве параметров при расчете метрик могут использоваться пропускная способность, стоимость передачи данных, задержка, счетчик транзитных узлов, загрузка, параметр MTU, стоимость пути и надежность. Часто используют упрощенное понятие — *метрика*.

Октет — 8 битов. В области сетевых технологий термин октет часто используется (чаще, чем байт) по причине того, что в некоторых структурах используют байт, который не равен 8 битам.

Пакет — логически сгруппированная единица информации, включающая заголовок, который содержит контрольную информацию, и (зачастую) пользовательские данные. Чаще всего о пакете говорят как о модуле передачи информации сетевого уровня. Термины *дейтаграмма*, *фрейм* и *сегмент* также описывают различные логические единицы информации на разных уровнях модели OSI и на разных технологических стадиях.

Переход (hop) — прохождение пакетом данных расстояния от одного сетевого узла, обычно маршрутизатора, к другому.

Подсеть. 1. В IP-сетях — часть сети с общим адресом подсети. Сеть делится на подсети произвольно сетевым администратором; при этом обеспечивается многоуровневая, иерархическая структура маршрутизации, в то же время нет необходимости в сложной адресации присоединенных сетей. 2. В сетях OSI — набор систем ES и IS, находящихся под контролем одного административного домена и использующих один протокол сетевого доступа.

Протокол внешнего шлюза (Exterior Gateway Protocol — EGP) — Internet-протокол, использующийся для обмена маршрутной информацией между автономными системами. Протокол граничного шлюза (Border Gateway Protocol — BGP) является наиболее распространенным протоколом класса EGP.

Протокол внутреннего шлюза (Interior Gateway Protocol — IGP) — Internet-протокол, использующийся для обмена маршрутной информацией внутри автономных систем. Примерами широко используемых протоколов класса IGP являются IGRP, OSPF и RIP.

Протокол маршрутизации внутреннего шлюза (Interior Gateway Routing Protocol — IGRP) — IGP-протокол, разработанный корпорацией Cisco для решения проблем маршрутизации в больших гетерогенных сетях.

Протокол маршрутизации — это протокол, реализующий маршрутизацию посредством использования определенного алгоритма поиска наилучшего пути. Примерами протоколов маршрутизации являются IGRP, OSPF и RIP.

Протокол маршрутной информации (Routing Information Protocol — RIP) — протокол IGP-типа, поставлявшийся с BSD UNIX-системами. Это наиболее широко распространенный протокол маршрутизации в локальных сетях. Протокол RIP использует в качестве метрики счетчик транзитных узлов.

Служба без установления соединения представляет собой механизм передачи данных без создания виртуального канала.

Служба с установлением соединения представляет собой механизм передачи данных, требующий установления виртуального канала.

Стек, или набор протоколов — это группа связанных, работающих совместно коммуникационных протоколов, обслуживающих взаимодействия на нескольких или всех семи уровнях модели OSI. Не все протоколы стека охватывают все уровни модели, и часто один протокол обслуживает одновременно несколько уровней. Типичным примером стека протоколов является набор TCP/IP.

Счетчик переходов (hop count) — это метрика маршрутизации, использующаяся для расчета расстояния между отправителем и получателем. Протокол RIP использует счетчик в качестве своей единственной метрики.

Таблица маршрутизации — таблица, хранящаяся в маршрутизаторах или некоторых других межсетевых устройствах и содержащая информацию о маршрутах до определенных сетей-получателей и, в некоторых случаях, связанные с ними метрики.

Широковещание — процесс, при котором информационный пакет рассылается всем узлам в сети. Получатель широковещательного пакета задается широковещательным адресом.

Широковещательный домен — это группа устройств, получающих широковещательные фреймы, отправленные одним из принадлежащих данной группе устройств. Обычно широковещательные домены ограничиваются маршрутизаторами (или в коммутируемых инфраструктурах посредством сетей VLAN), поскольку такие устройства не пересылают широковещательные фреймы.

Контрольные вопросы

Чтобы проверить, насколько хорошо вы усвоили темы и понятия, описанные в этой главе, ответьте на предлагаемые вопросы. Ответы на них приведены в приложении Б, “Ответы на контрольные вопросы”.

1. Из скольких битов состоит IP-адрес?

а) 16.

б) 32.

в) 64.

г) Ни один из перечисленных выше ответов не является правильным.

2. Каково максимальное значение любого из октетов в IP-адресе?

а) 28.

б) 255.

в) 256.

г) Ни один из перечисленных выше ответов не является правильным.

3. Какую роль играет номер сети в IP-адресе?

а) Он задает сеть, которой принадлежит узел.

б) Он идентифицирует компьютер в сети.

в) Он определяет, какой узел в подсети адресуется.

г) Он определяет, с какими сетями может взаимодействовать устройство.

4. Какую роль играет номер узла в IP-адресе?

а) Он идентифицирует компьютер в сети.

б) Он определяет, какой узел в подсети адресуется.

в) Он задает сеть, которой принадлежит узел.

г) Он указывает, с какими узлами может взаимодействовать устройство.

5. Какое число является десятичным эквивалентом двоичного числа 101101?

а) 32.

б) 35.

в) 45.

г) 44.

6. Какому числу в двоичной форме будет соответствовать десятичное число 192.5.34.11?

а) 11000000.00000101.00100010.00001011.

б) 11000101.01010111.00011000.10111000.

в) 01001011.10010011.00111001.00110111.

г) 11000000.00001010.01000010.00001011.

7. Какой комбинации в десятичной форме соответствует двоичный IP-адрес 11000000.00000101.00100010.00001011 ?

а) 190.4.34.11.

б) 192.4.34.10.

в) 192.4.32.11.

г) Ни один из вышеперечисленных.

8. Какая часть IP-адреса класса B 154.19.2.7 является номером сети?

а) 154.

б) 154.19.

в) 154.19.2.

г) 154.19.2.7.

9. Какая часть адреса 129.219.51.18 описывает сеть?

а) 129.219.

б) 129.

в) 14.1.

г) 1.

10. Какой из перечисленных ниже адресов является широковещательным в сети 123.10.0.0 с сетевой маской 255.255.0.0?

а) 123.255.255.255.

б) 123.10.255.255.

в) 123.13.0.0.

г) 123.1.1.1.

11. Сколько адресов узлов может быть использовано в сети класса C?

а) 253.

б) 254.

в) 255.

г) 256.

12. Какое минимальное число битов может быть заимствовано для формирования подсети?

а) 1.

б) 2.

в) 4.

г) Ни один из перечисленных выше ответов не является правильным.

13. Что является основной причиной для использования подсетей?

а) Уменьшение размеров домена коллизий.

б) Увеличение количества адресов узлов.

в) Уменьшение размеров широковещательного домена.

г) Ни один из перечисленных выше ответов не является правильным.

14. Сколько битов содержится в маске подсети?

а) 16.

б) 32.

в) 64.

г) Ни один из перечисленных выше ответов не является правильным.

15. Выполнив логическую операцию, которую совершает маршрутизатор над IP-адресом 121.8.2.5 и маской 255.0.0.0, вычислите адрес сети/подсети.

а) 121.8.1.0.

б) 121.8.0.0.

в) 121.8.2.0.

г) Ни один из перечисленных выше ответов не является правильным.

16. Сколько битов было заимствовано для создания подсетей в адресе класса C 197.15.22.31 с маской 255.255.255.224?

а) 1.

б) 2.

в) 3.

г) Ни один из перечисленных выше ответов не является правильным.

17. Выполнив логическую операцию, которую совершает маршрутизатор над IP-адресом 172.16.2.10 и маской 255.255.255.0, вычислите адрес подсети.

а) 172.0.0.0.

б) 172.16.0.0.

в) 172.16.2.0.

г) Ни один из перечисленных выше ответов не является правильным.

18. Какое выражение из перечисленных ниже наиболее точно описывает одну из функций третьего уровня — сетевого уровня модели OSI?

а) Он отвечает за надежное сетевое взаимодействие между узлами.

б) Он связан с физической адресацией и топологией сети.

в) Он определяет наилучший путь для потока данных, следующего через сеть.

г) Он обслуживает обмен информацией между уровнями представления разных систем.

19. Какая функция позволяет маршрутизаторам обнаруживать доступные маршруты до пункта назначения и выбирать наилучший из них при пересылке пакета?

а) Информационная связь.

б) Определение маршрута.

в) Протокол SDLC-интерфейса.

г) Frame Relay.

20. Каким образом сетевой уровень передает пакеты от отправителя получателю?

а) Посредством использования таблицы маршрутизации.

б) При помощи ARP-ответов.

в) С помощью запросов к серверу имен.

г) С помощью запросов к мосту.

21. Какие две части адреса сетевого уровня используют маршрутизаторы для передачи данных через сеть?

а) Адрес сети и адрес узла.

б) Адрес сети и MAC-адрес.

в) Адрес узла и MAC-адрес.

г) MAC-адрес и маску подсети.