24. Можно ли соединить транслирующим коммутатором сегменты, в которых установлено разное максимальное значение поля данных?
25. Имеется ли специфика в использовании мостов и коммутаторов? Приведите примеры, когда замена моста коммутатором не повышает производительности сети.
26. Почему недорогие коммутаторы, выполняющие ограниченное число функций, обычно работают по быстрому алгоритму обработки пакетов «на лету», а дорогие коммутаторы, с большим числом функций - по более медленному алгоритму буферизации пакетов?
27. Какая информация содержится в таблицах мостов/коммутаторов и маршрутизаторов?
28. Поясните определение: «Виртуальная локальная сеть - это домен распространения широковещательных сообщений».
29. В каких случаях появляется необходимость в создании виртуальных сегментов? Приведите примеры.

Сетевой уровень как средство построения больших сетей

5.1. Принципы объединения сетей на основе протоколов сетевого уровня

В стандартной модели взаимодействия открытых систем в функции сетевого уровня входит решение следующих задач:

- передача пакетов между конечными узлами в составных сетях;
- выбор маршрута передачи пакетов, наилучшего по некоторому критерию;
- согласование разных протоколов канального уровня, использующихся в отдельных подсетях одной составной сети.

Протоколы сетевого уровня реализуются, как правило, в виде программных модулей и выполняются на конечных узлах-компьютерах, называемых хостами, а также на промежуточных узлах - маршрутизаторах, называемых шлюзами. Функции маршрутизаторов могут выполнять как специализированные устройства, так и универсальные компьютеры с соответствующим программным обеспечением.

5.1.1. Ограничения мостов и коммутаторов

Создание сложной, структурированной сети, интегрирующей различные базовые технологии, может осуществляться и средствами канального уровня: для этого могут быть использованы некоторые типы мостов и коммутаторов. Мост или коммутатор разделяет сеть на сегменты, локализуя трафик внутри сегмента, что делает линии связи разделяемыми преимущественно между станциями данного сегмента. Тем самым сеть распадается на отдельные подсети, из которых могут быть построены составные сети достаточно крупных размеров.

Однако построение сложных сетей только на основе повторителей, мостов и коммутаторов имеет существенные ограничения и недостатки.

- Во-первых, в топологии получившейся сети должны *отсутствовать петли*. Действительно, мост/коммутатор может решать задачу доставки пакета адресату только тогда, когда между отправителем и получателем существует единственный путь. В то же время наличие избыточных связей, которые и образуют петли, часто необходимо для лучшей балансировки нагрузки, а также для повышения надежности сети за счет образования резервных путей.
- Во-вторых, логические сегменты сети, расположенные между мостами или коммутаторами, *слабо изолированы* друг от друга, а именно не защищены от так называемых широковещательных штормов. Если какая-либо станция посылает широковещательное сообщение, то это сообщение передается всем станциям всех логических сегментов сети. Защита от широковещательных штормов в сетях, построенных на основе мостов и коммутаторов, имеет количественный, а не качественный характер: администратор просто ограничивает количество широковещательных пакетов, которое разрешается генерировать некоторому узлу в единицу времени. Использование же механизма виртуальных сетей, реализованного во многих коммутаторах, хотя и позволяет достаточно гибко создавать изолированные по трафику группы станций, но при этом изолирует их полностью, так что узлы одной виртуальной сети не могут взаимодействовать с узлами другой виртуальной сети.
- В-третьих, в сетях, построенных на основе мостов и коммутаторов, достаточно сложно решается задача управления трафиком на основе значения данных, содержащихся в пакете. В таких сетях это возможно только с помощью пользовательских фильтров, для задания которых администратору приходится иметь дело с двоичным представлением содержимого пакетов.
- В-четвертых, реализация транспортной подсистемы только средствами физического и канального уровней, к которым относятся мосты и коммутаторы, приводит к недостаточно гибкой, одноуровневой системе адресации: в качестве адреса назначения используется MAC - адрес, жестко связанный с сетевым адаптером.
- Наконец, возможностью трансляции протоколов канального уровня обладают далеко не все типы мостов и коммутаторов, к тому же эти возможности ограничены. В частности, в объединяемых сетях должны совпадать максимально допустимые размеры полей данных в кадрах, так как мостами и коммутаторами не поддерживается функция фрагментации кадров. Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях - это привлечение средств более высокого, сетевого уровня.

5.1.2. Понятие internetworking

Основная идея введения сетевого уровня состоит в следующем. Сеть в общем случае рассматривается как совокупность нескольких сетей и называется составной сетью или интерсетью *(internetwork* или *internet).* Сети, входящие в составную сеть, называются подсетями *(subnet),* составляющими сетями или просто сетями (рис. 5.1).

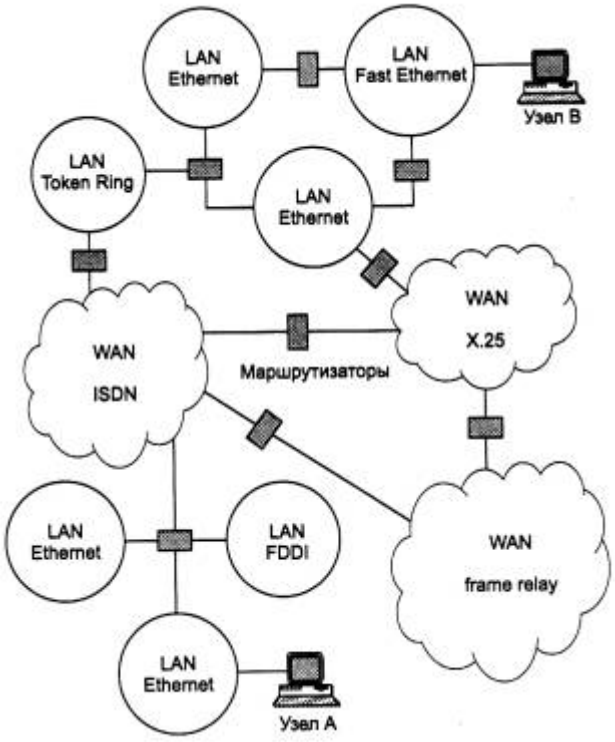


Рис. 5.1. Архитектура составной сети

Подсети соединяются между собой маршрутизаторами. Компонентами составной сети могут являться как локальные, так и глобальные сети. Внутренняя структура каждой сети на рисунке не показана, так как она не имеет значения при рассмотрении сетевого протокола. Все узлы в пределах одной подсети взаимодействуют, используя единую для них технологию. Так, в составную сеть, показанную на рисунке, входит несколько сетей разных технологий: локальные сети Ethernet, Fast Ethernet, Token Ring, FDDI и глобальные сети frame relay, X.25, ISDN. Каждая из этих технологий достаточна для того, чтобы организовать взаимодействие всех узлов в своей подсети, но не способна построить информационную связь между произвольно выбранными узлами, принадлежащими разным подсетям, например между узлом А и узлом В на рис. 5.1. Следовательно, для организации взаимодействия между любой произвольной парой узлов этой «большой» составной сети требуются дополнительные средства. Такие средства и предоставляет сетевой уровень.

Сетевой уровень выступает в качестве координатора, организующего работу всех подсетей, лежащих на пути продвижения пакета по составной сети. Для перемещения данных в пределах подсетей сетевой уровень обращается к используемым в этих подсетях технологиям.

Хотя многие технологии локальных сетей (Ethernet, Token Ring, FDDI, Fast Ethernet и др.) используют одну и ту же систему адресации узлов на основе MAC - адресов, существует немало технологий (X.25, ATM, frame relay), в которых применяются другие схемы

адресации. Адреса, присвоенные узлам в соответствии с технологиями подсетей, называют локальными. Чтобы сетевой уровень мог выполнить свою задачу, ему необходима собственная система адресации, не зависящая от способов адресации узлов в отдельных подсетях, которая позволила бы на сетевом уровне универсальным и однозначным способами идентифицировать любой узел составной сети.

Естественным способом формирования сетевого адреса является уникальная нумерация всех подсетей составной сети и нумерация всех узлов в пределах каждой подсети. Таким образом, сетевой адрес представляет собой пару: номер сети (подсети) и номер узла.

В качестве номера узла может выступать либо локальный адрес этого узла (такая схема принята в стеке IPX/SPX), либо некоторое число, никак не связанное с локальной технологией, которое однозначно идентифицирует узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, что ограничивает его применение. Например, сетевые адреса IPX/SPX рассчитаны на работу в составных сетях, объединяющих сети, в которых используются только MAC - адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP. И в том и другом случае каждый узел составной сети имеет наряду со своим локальным адресом еще один - универсальный сетевой адрес.

Данные, которые поступают на сетевой уровень и которые необходимо передать через составную сеть, снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют пакет. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в объединенную сеть, и несет наряду с другой служебной информацией данные о номере сети, которой предназначается этот пакет. Сетевой уровень определяет маршрут и перемещает пакет между подсетями.

При передаче пакета из одной подсети в другую пакет сетевого уровня, инкапсулированный в прибывший канальный кадр первой подсети, освобождается от заголовков этого кадра и окружается заголовками кадра канального уровня следующей подсети. Информацией, на основе которой делается эта замена, являются служебные поля пакета сетевого уровня. В поле адреса назначения нового кадра указывается локальный адрес следующего маршрутизатора.

ПРИМЕЧАНИЕ Если в подсети доставка данных осуществляется средствами канального и физического уровней (как, например, в стандартных локальных сетях), то пакеты сетевого уровня упаковываются в кадры канального уровня. Если же в какой-либо подсети для транспортировки сообщений используется технология, основанная на стеках с большим числом уровней, то пакеты сетевого уровня упаковываются в блоки передаваемых данных самого высокого уровня подсети.

Если проводить аналогию между взаимодействием разнородных сетей и перепиской людей из разных стран, то сетевая информация - это общепринятый индекс страны, добавленный к адресу письма, написанному на одном из сотни языков земного шара, например на санскрите. И даже если это письмо должно пройти через множество стран, почтовые работники которых не знают санскрита, понятный им индекс страны-адресата

подскажет, через какие промежуточные страны лучше передать письмо, чтобы оно кратчайшим путем попало в Индию. А уже там работники местных почтовых отделений смогут прочитать точный адрес, указывающий город, улицу, дом и индивидуума, и доставить письмо адресату, так как адрес написан на языке и в форме, принятой в данной стране.

Основным полем заголовка сетевого уровня является номер сети-адресата. В рассмотренных нами ранее протоколах локальных сетей такого поля в кадрах предусмотрено не было - предполагалось, что все узлы принадлежат одной сети. Явная нумерация сетей позволяет протоколам сетевого уровня составлять точную карту межсетевых связей и выбирать рациональные маршруты при любой их топологии, в том числе альтернативные маршруты, если они имеются, что не умеют делать мосты и коммутаторы.

Кроме номера сети заголовок сетевого уровня должен содержать и другую информацию, необходимую для успешного перехода пакета из сети одного типа в сеть другого типа. К такой информации может относиться, например:

- номер фрагмента пакета, необходимый для успешного проведения операций сборки-разборки фрагментов при соединении сетей с разными максимальными размерами пакетов;
- время жизни пакета, указывающее, как долго он путешествует по интерсети, это время может использоваться для уничтожения «заблудившихся» пакетов;
- качество услуги - критерий выбора маршрута при межсетевых передачах - например, узел-отправитель может потребовать передать пакет с максимальной надежностью, возможно, в ущерб времени доставки.

Когда две или более сети организуют совместную транспортную службу, то такой режим взаимодействия обычно называют межсетевым взаимодействием (internetworking).

5.1.3. Принципы маршрутизации

Важнейшей задачей сетевого уровня является маршрутизация - передача пакетов между двумя конечными узлами в составной сети.

Рассмотрим принципы маршрутизации на примере составной сети, изображенной на рис. 5.2. В этой сети 20 маршрутизаторов объединяют 18 сетей в общую сеть; S1, S2, ... , S20 - это номера сетей. Маршрутизаторы имеют по нескольку портов (по крайней мере, по два), к которым присоединяются сети. Каждый порт маршрутизатора можно рассматривать как отдельный узел сети: он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор под номером 1 имеет три порта, к которым подключены сети S1, S2, S3. На рисунке сетевые адреса этих портов обозначены как M1(1), Ml (2) и M1(3). Порт M1(1) имеет локальный адрес в сети с номером S1, порт Ml (2) - в сети S2, а порт M1(3) - в сети S3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет ни отдельного сетевого адреса, ни какого-либо локального адреса.

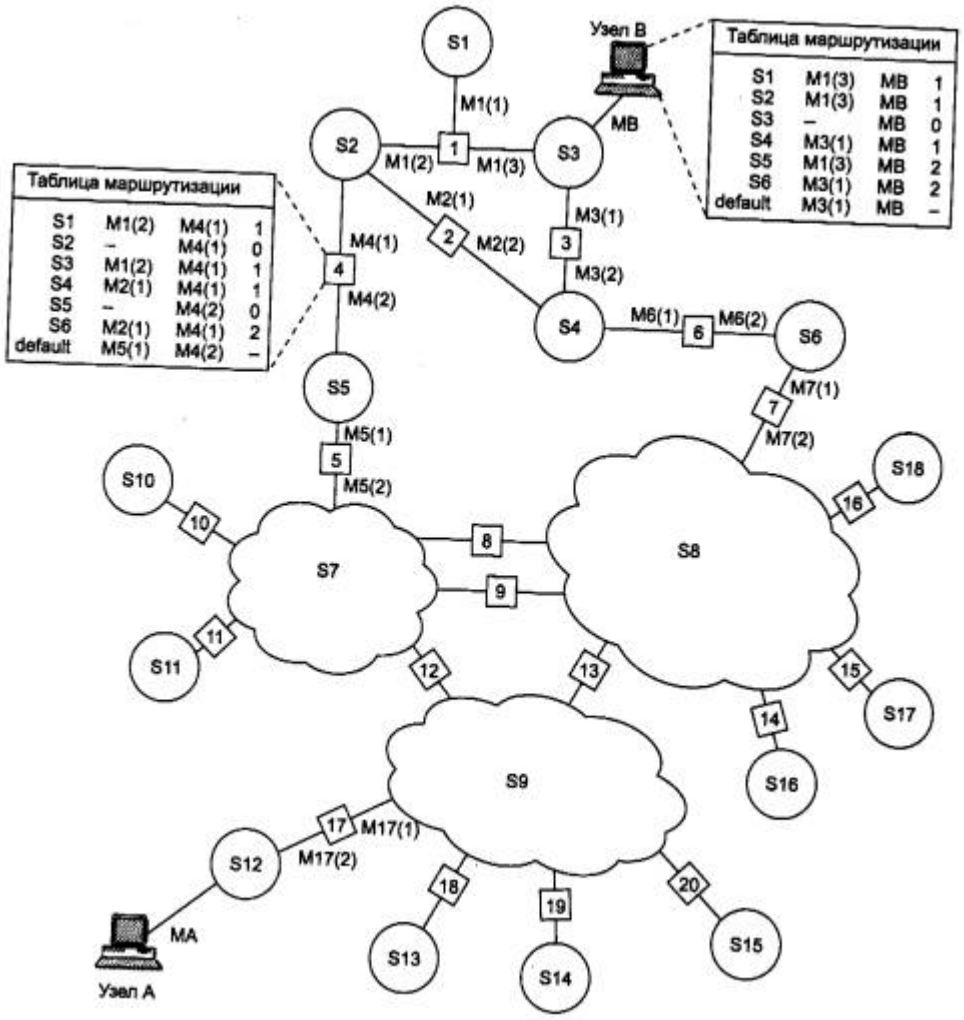


Рис. 5.2. Принципы маршрутизации в составной сети

ПРИМЕЧАНИЕ Если маршрутизатор имеет блок управления (например, SNMP-управления), то этот блок имеет собственные локальный и сетевой адреса, по которым к нему обращается центральная станция управления, находящаяся где-то в составной сети.

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Маршрут - это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до пункта назначения. Так, пакет, отправленный из узла А в узел В, может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17,13, 7, 6 и 3. Нетрудно найти еще несколько маршрутов между узлами А и В.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. Обычно в качестве критерия выступает задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута для

последовательности пакетов. Часто также используется весьма простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (хопов).

Чтобы по адресу сети назначения можно было бы выбрать рациональный маршрут дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализируют специальную информационную структуру, которая называется таблицей маршрутизации. Используя условные обозначения для сетевых адресов маршрутизаторов и номеров сетей в том виде, как они приведены на рис. 5.2, посмотрим, как могла бы выглядеть таблица маршрутизации, например, в маршрутизаторе 4 (табл. 5.1).

Таблица 5.1. Таблица маршрутизации маршрутизатора 4

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
S1	M1(2)	M4(1)	1
S2	—	M4(1)	0 (подсоединена)
S3	M1(2)	M4(1)	1
S4	M2(1)	M4(1)	1
S5	—	M4(2)	0 (подсоединена)
S6	M2(1)	M4(1)	2
Default	M5(1)	M4(2)	—

ПРИМЕЧАНИЕ Таблица 5.1 значительно упрощена по сравнению с реальными таблицами, например, отсутствуют столбцы с масками, признаками состояния маршрута, временем, в течение которого действительны записи данной таблицы (их применение будет рассмотрено позже). Кроме того, как уже было сказано, здесь указаны адреса сетей условного формата, не соответствующие какому-либо определенному сетевому протоколу. Тем не менее эта таблица содержит основные поля, имеющиеся в реальных таблицах при использовании конкретных сетевых протоколов, таких как IP, IPX или X.25.

В первом столбце таблицы перечисляются номера сетей, входящих в интерсеть. В каждой строке таблицы следом за номером сети указывается сетевой адрес следующего маршрутизатора (более точно, сетевой адрес соответствующего порта следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к сети с данным номером по рациональному маршруту.

Когда на маршрутизатор поступает новый пакет, номер сети назначения, извлеченный из поступившего кадра, последовательно сравнивается с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает, на какой ближайший маршрутизатор следует направить пакет. Например, если на какой-либо порт маршрутизатора 4 поступает пакет, адресованный в сеть S6, то из таблицы маршрутизации следует, что адрес следующего маршрутизатора - M2(1), то есть очередным этапом движения данного пакета будет движение к порту 1 маршрутизатора 2.

Поскольку пакет может быть адресован в любую сеть составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо всех сетях, входящих в составную сеть. Но при таком подходе в случае крупной сети объем таблиц маршрутизации может оказаться очень большим, что повлияет на время ее просмотра, потребует много места для хранения и т. п. Поэтому на практике число записей в таблице маршрутизации стараются уменьшить за счет использования специальной записи - *«маршрутизатор по умолчанию» (default)*. Действительно, если принять во внимание топологию составной сети, то в таблицах маршрутизаторов, находящихся на периферии составной сети, достаточно записать номера сетей, непосредственно подсоединенных к данному маршрутизатору или расположенных поблизости, на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию, а вместо номера сети в соответствующей строке помещается особая запись, например default. В нашем примере таким маршрутизатором по умолчанию для сети S5 является маршрутизатор 5, точнее его порт M5(1). Это означает, что путь из сети S5 почти ко всем сетям большой составной сети пролегает через этот порт маршрутизатора.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации. Еще раз подчеркнем, что каждый порт идентифицируется собственным сетевым адресом.

Некоторые реализации сетевых протоколов допускают наличие в таблице маршрутизации сразу нескольких строк, соответствующих одному и тому же адресу сети назначения. В этом случае при выборе маршрута принимается во внимание столбец «Расстояние до сети назначения». При этом под расстоянием понимается любая метрика, используемая в соответствии с заданным в сетевом пакете критерием (часто называемым классом сервиса). Расстояние может измеряться хопами, временем прохождения пакета по линиям связи, какой-либо характеристикой надежности линий связи на данном маршруте или другой величиной, отражающей качество данного маршрута по отношению к заданному критерию. Если маршрутизатор поддерживает несколько классов сервиса пакетов, то таблица маршрутов составляется и применяется отдельно для каждого вида сервиса (критерия выбора маршрута).

В табл. 5.1 расстояние между сетями измерялось хопами. Расстояние для сетей, непосредственно подключенных к портам маршрутизатора, здесь принимается равным 0, однако в некоторых реализациях отсчет расстояний начинается с 1.

Наличие нескольких маршрутов к одному узлу делают возможным передачу трафика к этому узлу параллельно по нескольким каналам связи, это повышает пропускную способность и надежность сети.

Задачу маршрутизации решают не только промежуточные узлы - маршрутизаторы, но и конечные узлы - компьютеры. Средства сетевого уровня, установленные на конечном узле, при обработке пакета должны, прежде всего, определить, направляется ли он в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, то для данного пакета не требуется решать задачу маршрутизации. Если же номера сетей отправления и назначения не совпадают, то маршрутизация нужна. Таблицы маршрутизации конечных узлов полностью аналогичны таблицам маршрутизации, хранящимся на маршрутизаторах.

Обратимся снова к сети, изображенной на рис. 5.2. Таблица маршрутизации для конечного узла B могла бы выглядеть следующим образом (табл. 5.2). Здесь MB - сетевой адрес порта компьютера B. На основании этой таблицы конечный узел B выбирает, на какой из двух имеющихся в локальной сети S3 маршрутизаторов следует посылать тот или иной пакет.

Таблица 5.2. Таблица маршрутизации конечного узла B

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
S1	M1(3)	MB	1
S2	M1(3)	MB	1
S3	—	MB	0
S4	M3(1)	MB	1
S5	M1(3)	MB	2
S6	M3(1)	MB	2
Default	M3(1)	MB	—

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своем распоряжении таблицу маршрутизации, ее объем обычно незначителен, что объясняется периферийным расположением всех конечных узлов. Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант - единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда перед конечным узлом стоит проблема их выбора, задание маршрута по умолчанию часто используется в компьютерах для сокращения объема их таблицы маршрутизации.

Ниже помещена таблица маршрутизации другого конечного узла составной сети - узла A (табл. 5.3). Компактный вид таблицы маршрутизации отражает тот факт, что все пакеты, направляемые из узла A, либо не выходят за пределы сети S12, либо непременно проходят через порт 1 маршрутизатора 17. Этот маршрутизатор и определен в таблице маршрутизации в качестве маршрутизатора по умолчанию.

Таблица 5.3. Таблица маршрутизации конечного узла A

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
S12	—	MA	0
Default	M17(1)	MA	—

Еще одним отличием работы маршрутизатора и конечного узла при выборе маршрута является способ построения таблицы маршрутизации. Если маршрутизаторы обычно автоматически создают таблицы маршрутизации, обмениваясь служебной информацией, то для конечных узлов таблицы маршрутизации часто создаются вручную администраторами и хранятся в виде постоянных файлов на дисках.

5.1.4. Протоколы маршрутизации

Задача маршрутизации решается на основе анализа таблиц маршрутизации, размещенных во всех маршрутизаторах и конечных узлах сети. Каким же образом происходит формирование этих таблиц? Какими средствами обеспечивается адекватность содержащейся в них информации постоянно изменяющейся структуре сети? Основная работа по созданию таблиц маршрутизации выполняется автоматически, но и возможность вручную скорректировать или дополнить таблицу тоже, как правило, предусматривается.

Для автоматического построения таблиц маршрутизации маршрутизаторы обмениваются информацией о топологии составной сети в соответствии со специальным служебным протоколом. Протоколы этого типа называются протоколами маршрутизации (или маршрутизирующими протоколами). Протоколы маршрутизации (например, RIP, OSPF, NLSP) следует отличать от собственно сетевых протоколов (например, IP, IPX). И те и другие выполняют функции сетевого уровня модели OSI - участвуют в доставке пакетов адресату через разнородную составную сеть. Но в то время как первые собирают и передают по сети чисто служебную информацию, вторые предназначены для передачи пользовательских данных, как это делают протоколы канального уровня. Протоколы маршрутизации используют сетевые протоколы как транспортное средство. При обмене маршрутной информацией пакеты протокола маршрутизации помещаются в поле данных пакетов сетевого уровня или даже транспортного уровня, поэтому с точки зрения вложенности пакетов протоколы маршрутизации формально следовало бы отнести к более высокому уровню, чем сетевой.

В том, что маршрутизаторы для принятия решения о продвижении пакета обращаются к адресным таблицам, можно увидеть их некоторое сходство с мостами и коммутаторами. Однако природа используемых ими адресных таблиц сильно различается. Вместо MAC - адресов в таблицах маршрутизации указываются номера сетей, которые соединяются в интерсеть. Другим отличием таблиц маршрутизации от адресных таблиц мостов является способ их создания. В то время как мост строит таблицу, пассивно наблюдая за проходящими через него информационными кадрами, посылаемыми конечными узлами сети друг другу, маршрутизаторы по своей инициативе обмениваются специальными служебными пакетами, сообщая соседям об известных им сетях в интерсети, маршрутизаторах и о связях этих сетей с маршрутизаторами. Обычно учитывается не только топология связей, но и их пропускная способность и состояние. Это позволяет маршрутизаторам быстрее адаптироваться к изменениям конфигурации сети, а также правильно передавать пакеты в сетях с произвольной топологией, допускающей наличие замкнутых контуров.

С помощью протоколов маршрутизации маршрутизаторы составляют карту связей сети той или иной степени подробности. На основании этой информации для каждого номера сети принимается решение о том, какому следующему маршрутизатору надо передавать пакеты, направляемые в эту сеть, чтобы маршрут оказался рациональным. Результаты этих решений заносятся в таблицу маршрутизации. При изменении конфигурации сети некоторые записи в таблице становятся недействительными. В таких случаях пакеты, отправленные по ложным маршрутам, могут зацикливаться и теряться. От того, насколько быстро протокол маршрутизации приводит в соответствие содержимое таблицы реальному состоянию сети, зависит качество работы всей сети.

Протоколы маршрутизации могут быть построены на основе разных алгоритмов, отличающихся способами построения таблиц маршрутизации, способами выбора наилучшего маршрута и другими особенностями своей работы.

Во всех описанных выше примерах при выборе рационального маршрута определялся только следующий (ближайший) маршрутизатор, а не вся последовательность маршрутизаторов от начального до конечного узла. В соответствии с этим подходом маршрутизация выполняется по распределенной схеме - каждый маршрутизатор ответственен за выбор только одного шага маршрута, а окончательный маршрут складывается в результате работы всех маршрутизаторов, через которые проходит данный пакет. Такие алгоритмы маршрутизации называются *одношаговыми*.

Существует и прямо противоположный, многошаговый подход - *маршрутизация от источника (Source Routing).* В соответствии с ним узел-источник задает в отправляемом в сеть пакете полный маршрут его следования через все промежуточные маршрутизаторы. При использовании многошаговой маршрутизации нет необходимости строить и анализировать таблицы маршрутизации. Это ускоряет прохождение пакета по сети, разгружает маршрутизаторы, но при этом большая нагрузка ложится на конечные узлы. Эта схема в вычислительных сетях применяется сегодня гораздо реже, чем схема распределенной одношаговой маршрутизации. Однако в новой версии протокола IP наряду с классической одношаговой маршрутизацией будет разрешена и маршрутизация от источника.

Одношаговые алгоритмы в зависимости от способа формирования таблиц маршрутизации делятся на три класса:

- алгоритмы фиксированной (или статической) маршрутизации;
- алгоритмы простой маршрутизации;
- алгоритмы адаптивной (или динамической) маршрутизации.

В алгоритмах *фиксированной маршрутизации* все записи в таблице маршрутизации являются статическими. Администратор сети сам решает, на какие маршрутизаторы надо передавать пакеты с теми или иными адресами, и вручную (например, с помощью утилиты route ОС Unix или Windows NT) заносит соответствующие записи в таблицу маршрутизации. Таблица, как правило, создается в процессе загрузки, в дальнейшем она используется без изменений до тех пор, пока ее содержимое не будет отредактировано вручную. Такие исправления могут понадобиться, например, если в сети отказывает какой-либо маршрутизатор и его функции возлагаются на другой маршрутизатор. Различают одномаршрутные таблицы, в которых для каждого адресата задан один путь, и многомаршрутные таблицы, определяющие несколько альтернативных путей для каждого адресата. В многомаршрутных таблицах должно быть задано правило выбора одного из маршрутов. Чаще всего один путь является основным, а остальные - резервными. Понятно, что алгоритм фиксированной маршрутизации с его ручным способом формирования таблиц маршрутизации приемлем только в небольших сетях с простой топологией. Однако этот алгоритм может быть эффективно использован и для работы на магистралях крупных сетей, так как сама магистраль может иметь простую структуру с очевидными наилучшими путями следования пакетов в подсети, присоединенные к магистрали.

В алгоритмах *простой маршрутизации* таблица маршрутизации либо вовсе не используется, либо строится без участия протоколов маршрутизации. Выделяют три типа простой маршрутизации:

- *случайная маршрутизация*, когда прибывший пакет посылается в первом попавшем случайном направлении, кроме исходного;

- *лавинная маршрутизация*, когда пакет широковещательно посылается по всем возможным направлениям, кроме исходного (аналогично обработке мостами кадров с неизвестным адресом);
- *маршрутизация по предыдущему опыту*, когда выбор маршрута осуществляется по таблице, но таблица строится по принципу моста путем анализа адресных полей пакетов, появляющихся на входных портах.

Самыми распространенными являются алгоритмы *адаптивной (или динамической) маршрутизации*. Эти алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети. Протоколы, построенные на основе адаптивных алгоритмов, позволяют всем маршрутизаторам собирать информацию о топологии связей в сети, оперативно отрабатывая все изменения конфигурации связей. В таблицах маршрутизации при адаптивной маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют *временем жизни маршрута (Time To Live, TTL)*.

Адаптивные алгоритмы обычно имеют распределенный характер, который выражается в том, что в сети отсутствуют какие-либо выделенные маршрутизаторы, которые собирали бы и обобщали топологическую информацию: эта работа распределена между всеми маршрутизаторами.

ПРИМЕЧАНИЕ В последнее время наметилась тенденция использовать так называемые серверы маршрутов. Сервер маршрутов собирает маршрутную информацию, а затем раздает ее по запросам маршрутизаторам, которые освобождаются в этом случае от функции создания таблиц маршрутизации, либо создают только части этих таблиц. Появились специальные протоколы взаимодействия маршрутизаторов с серверами маршрутов, например Next Hop Resolution Protocol (NHRP).

Адаптивные алгоритмы маршрутизации должны отвечать нескольким важным требованиям. Во-первых, они должны обеспечивать, если не оптимальность, то хотя бы рациональность маршрута. Во-вторых, алгоритмы должны быть достаточно простыми, чтобы при их реализации не тратилось слишком много сетевых ресурсов, в частности они не должны требовать слишком большого объема вычислений или порождать интенсивный служебный трафик. И наконец, алгоритмы маршрутизации должны обладать свойством сходимости, то есть всегда приводить к однозначному результату за приемлемое время.

Адаптивные протоколы обмена маршрутной информацией, применяемые в настоящее время в вычислительных сетях, в свою очередь делятся на две группы, каждая из которых связана с одним из следующих типов алгоритмов:

- дистанционно-векторные алгоритмы (Distance Vector Algorithms, DVA);
- алгоритмы состояния связей (Link State Algorithms, LSA).

В алгоритмах *дистанционно-векторного типа* каждый маршрутизатор периодически и широковещательно рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей. Под расстоянием обычно понимается число хопов. Возможна и другая метрика, учитывающая не только

число промежуточных маршрутизаторов, но и время прохождения пакетов по сети между соседними маршрутизаторами. При получении вектора от соседа маршрутизатор наращивает расстояния до указанных в векторе сетей на расстояние до данного соседа. Получив вектор от соседнего маршрутизатора, каждый маршрутизатор добавляет к нему информацию об известных ему других сетях, о которых он узнал непосредственно (если они подключены к его портам) или из аналогичных объявлений других маршрутизаторов, а затем снова рассылает новое значение вектора по сети. В конце концов, каждый маршрутизатор узнает информацию обо всех имеющихся в интерсети сетях и о расстоянии до них через соседние маршрутизаторы.

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях, В больших сетях они засоряют линии связи интенсивным широковещательным трафиком, к тому же изменения конфигурации могут отрабатываться по этому алгоритму не всегда корректно, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только обобщенной информацией - вектором дистанций, к тому же полученной через посредников. Работа маршрутизатора в соответствии с дистанционно-векторным протоколом напоминает работу моста, так как точной топологической картины сети такой маршрутизатор не имеет.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP, который распространен в двух версиях - RIP IP, работающий с протоколом IP, и RIP IPX, работающий с протоколом IPX.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на основании одинаковых графов, что делает процесс маршрутизации более устойчивым к изменениям конфигурации. «Широковещательная» рассылка (то есть передача пакета всем непосредственным соседям маршрутизатора) используется здесь только при изменениях состояния связей, что происходит в надежных сетях не так часто. Вершинами графа являются как маршрутизаторы, так и объединяемые ими сети. Распространяемая по сети информация состоит из описания связей различных типов: маршрутизатор - маршрутизатор, маршрутизатор - сеть,

Чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами HELLO со своими ближайшими соседями. Этот служебный трафик также засоряет сеть, но не в такой степени как, например, RIP-пакеты, так как пакеты HELLO имеют намного меньший объем.

Протоколами, основанными на алгоритме состояния связей, являются протоколы IS-IS (Intermediate System to Intermediate System) стека OSI, OSPF (Open Shortest Path First) стека TCP/IP и недавно реализованный протокол NLSP стека Novell.

5.1.5. Функции маршрутизатора

Основная функция маршрутизатора - чтение заголовков пакетов сетевых протоколов, принимаемых и буферизуемых по каждому порту (например, IPX, IP, AppleTalk или DECnet), и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу, включающему, как правило, номер сети и номер узла.

Функции маршрутизатора могут быть разбиты на 3 группы в соответствии с уровнями модели OSI (рис. 5.3).

Рис. 5.3. Функциональная модель маршрутизатора

Уровень интерфейсов

На нижнем уровне маршрутизатор, как и любое устройство, подключенное к сети, обеспечивает физический интерфейс со средой передачи, включая согласование уровней электрических сигналов, линейное и логическое кодирование, оснащение определенным типом разъема. В разных моделях маршрутизаторов часто предусматриваются различные наборы физических интерфейсов, представляющих собой комбинацию портов для подсоединения локальных и глобальных сетей. С каждым интерфейсом для подключения локальной сети неразрывно связан определенный протокол канального уровня - например, Ethernet, Token Ring, FDDI. Интерфейсы для присоединения к глобальным сетям чаще всего определяют только некоторый стандарт физического уровня, над которым в маршрутизаторе могут работать различные протоколы канального уровня. Например, глобальный порт может поддерживать интерфейс V.35, над которым могут работать протоколы канального уровня: LAP-B (используемый в сетях X.25), LAP-F (используемый в сетях frame relay), LAP-D (используемый в сетях ISDN). Разница между интерфейсами локальных и глобальных сетей объясняется тем, что технологии локальных сетей работают по собственным стандартам физического уровня, которые не могут, как правило, использоваться в других технологиях, поэтому интерфейс для локальной сети представляет собой сочетание физического и канального уровней и носит название по имени соответствующей технологии - например, интерфейс Ethernet.

Интерфейсы маршрутизатора выполняют полный набор функций физического и канального уровней по передаче кадра, включая получение доступа к среде (если это

необходимо), формирование битовых сигналов, прием кадра, подсчет его контрольной суммы и передачу поля данных кадра верхнему уровню, в случае если контрольная сумма имеет корректное значение.

ПРИМЕЧАНИЕ Как и любой конечный узел, каждый порт маршрутизатора имеет собственный аппаратный адрес (в локальных сетях MAC - адрес), по которому ему и направляются кадры, требующие маршрутизации, другими узлами сети.

Перечень физических интерфейсов, которые поддерживает та или иная модель маршрутизатора, является его важнейшей потребительской характеристикой. Маршрутизатор должен поддерживать все протоколы канального и физического уровней, используемые в каждой из сетей, к которым он будет непосредственно присоединен. На рис. 5.3 показана функциональная модель маршрутизатора с четырьмя портами, реализующими следующие физические интерфейсы: 10Base-T и 10Base-2 для двух портов Ethernet, UTP для Token Ring и V.35, над которым могут работать протоколы LAP-B, LAP-D или LAP-F, обеспечивая подключение к сетям X.25, ISDN или frame relay.

Кадры, которые поступают на порты маршрутизатора, после обработки соответствующими протоколами физического и канального уровней, освобождаются от заголовков канального уровня. Извлеченные из поля данных кадра пакеты передаются модулю сетевого протокола.

Уровень сетевого протокола

Сетевой протокол в свою очередь извлекает из пакета заголовок сетевого уровня и анализирует содержимое его полей. Прежде всего проверяется контрольная сумма, и если пакет пришел поврежденным, то он отбрасывается. Выполняется проверка, не превысило ли время, которое провел пакет в сети (время жизни пакета), допустимой величины. Если превысило - то пакет также отбрасывается. На этом этапе вносятся корректировки в содержимое некоторых полей, например, наращивается время жизни пакета, пересчитывается контрольная сумма.

На сетевом уровне выполняется одна из важнейших функций маршрутизатора - фильтрация трафика. Маршрутизатор, обладая более высоким интеллектом, нежели мосты и коммутаторы, позволяет задавать и может отрабатывать значительно более сложные правила фильтрации. Пакет сетевого уровня, находящийся в поле данных кадра, для мостов/коммутаторов представляется неструктурированной двоичной последовательностью. Маршрутизаторы же, программное обеспечение которых содержит модуль сетевого протокола, способны производить разбор и анализ отдельных полей пакета. Они оснащаются развитыми средствами пользовательского интерфейса, которые позволяют администратору без особых усилий задавать сложные правила фильтрации. Они, например, могут запретить прохождение в корпоративную сеть всех пакетов, кроме пакетов, поступающих из подсетей этого же предприятия. Фильтрация в данном случае производится по сетевым адресам, и все пакеты, адреса которых не входят в разрешенный диапазон, отбрасываются. Маршрутизаторы, как правило, также могут анализировать структуру сообщений транспортного уровня, поэтому фильтры могут не пропускать в сеть

сообщения определенных прикладных служб, например службы tehet, анализируя поле типа протокола в транспортном сообщении.

В случае если интенсивность поступления пакетов выше интенсивности, с которой они обрабатываются, пакеты могут образовать очередь. Программное обеспечение маршрутизатора может реализовать различные дисциплины обслуживания очередей пакетов: в порядке поступления по принципу «первый пришел - первым обслужен» (First Input First Output, FIFO), случайное раннее обнаружение, когда обслуживание идет по правилу FIFO, но при достижении длиной очереди некоторого порогового значения вновь поступающие пакеты отбрасываются (Random Early Detection, RED), а также различные варианты приоритетного обслуживания.

К сетевому уровню относится основная функция маршрутизатора - определение маршрута пакета. По номеру сети, извлеченному из заголовка пакета, модуль сетевого протокола находит в таблице маршрутизации строку, содержащую сетевой адрес следующего маршрутизатора, и номер порта, на который нужно передать данный пакет, чтобы он двигался в правильном направлении. Если в таблице отсутствует запись о сети назначения пакета и к тому же нет записи о маршрутизаторе по умолчанию, то данный пакет отбрасывается.

Перед тем как передать сетевой адрес следующего маршрутизатора на канальный уровень, необходимо преобразовать его в локальный адрес той технологии, которая используется в сети, содержащей следующий маршрутизатор. Для этого сетевой протокол обращается к *протоколу разрешения адресов*. Протоколы этого типа устанавливают соответствие между сетевыми и локальными адресами либо на основании заранее составленных таблиц, либо путем рассылки широковещательных запросов. Таблица соответствия локальных адресов сетевым адресам строится отдельно для каждого сетевого интерфейса. Протоколы разрешения адресов занимают промежуточное положение между сетевым и канальным уровнями.

С сетевого уровня пакет, локальный адрес следующего маршрутизатора и номер порта маршрутизатора передаются вниз, канальному уровню. На основании указанного номера порта осуществляется коммутация с одним из интерфейсов маршрутизатора, средствами которого выполняется упаковка пакета в кадр соответствующего формата. В поле адреса назначения заголовка кадра помещается локальный адрес следующего маршрутизатора. Готовый кадр отправляется в сеть.

Уровень протоколов маршрутизации

Сетевые протоколы активно используют в своей работе таблицу маршрутизации, но ни ее построением, ни поддержанием ее содержимого не занимаются. Эти функции выполняют протоколы маршрутизации. На основании этих протоколов маршрутизаторы обмениваются информацией о топологии сети, а затем анализируют полученные сведения, определяя наилучшие по тем или иным критериям маршруты. Результаты анализа и составляют содержимое таблиц маршрутизации.

Помимо перечисленных выше функций, на маршрутизаторы могут быть возложены и другие обязанности, например операции, связанные с фрагментацией. Более детально работа маршрутизаторов будет описана при рассмотрении конкретных протоколов сетевого уровня.

5.1.6. Реализация межсетевого взаимодействия средствами TCP/IP

В настоящее время стек TCP/IP является самым популярным средством организации составных сетей. На рис. 5.4 показана доля, которую составляет тот или иной стек протоколов в общемировой инсталляционной сетевой базе. До 1996 года бесспорным лидером был стек IPX/SPX компании Novell, но затем картина резко изменилась - стек TCP/IP по темпам роста числа установок намного стал опережать другие стеки, а с 1998 года вышел в лидеры и в абсолютном выражении. Именно поэтому дальнейшее изучение функций сетевого уровня будет проводиться на примере стека TCP/IP.

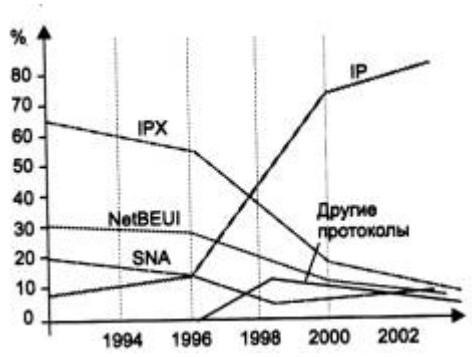


Рис. 5.4. Стек TCP/IP становится основным средством построения составных сетей

Многоуровневая структура стека TCP/IP

В стеке TCP/IP определены 4 уровня (рис. 5.5). Каждый из этих уровней несет на себе некоторую нагрузку по решению основной задачи - организации надежной и производительной работы составной сети, части которой построены на основе разных сетевых технологий.

Уровень I	Прикладной уровень
Уровень II	Основной (транспортный) уровень
Уровень III	Уровень межсетевого взаимодействия
Уровень IV	Уровень сетевых интерфейсов

Рис. 5.5. Многоуровневая архитектура стека TCP/IP

Уровень межсетевого взаимодействия

Стержнем всей архитектуры является *уровень межсетевого взаимодействия*, который реализует концепцию передачи пакетов в режиме без установления соединений, то есть дейтаграммным способом. Именно этот уровень обеспечивает возможность перемещения пакетов по сети, используя тот маршрут, который в данный момент является наиболее рациональным. Этот уровень также называют уровнем internet, указывая тем самым на основную его функцию - передачу данных через составную сеть.

Основным протоколом сетевого уровня (в терминах модели OSI) в стеке является протокол IP (Internet Protocol). Этот протокол изначально проектировался как протокол передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, объединенных как локальными, так и глобальными связями. Поэтому протокол IP хорошо работает в сетях со сложной топологией, рационально используя наличие в них подсистем

и экономно расходуя пропускную способность низкоскоростных линий связи. Так как протокол IP является дейтаграммным протоколом, он не гарантирует доставку пакетов до узла назначения, но старается это сделать.

К уровню межсетевого взаимодействия относятся и все протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как протоколы сбора маршрутной информации RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), а также протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol). Последний протокол предназначен для обмена информацией об ошибках между маршрутизаторами сети и узлом-источником пакета. С помощью специальных пакетов ICMP сообщает о невозможности доставки пакета, о превышении времени жизни или продолжительности сборки пакета из фрагментов, об аномальных величинах параметров, об изменении маршрута пересылки и типа обслуживания, о состоянии системы и т. п.

Основной уровень

Поскольку на сетевом уровне не устанавливаются соединения, то нет никаких гарантий, что все пакеты будут доставлены в место назначения целыми и невредимыми или придут в том же порядке, в котором они были отправлены. Эту задачу -обеспечение надежной информационной связи между двумя конечными узлами -решает *основной уровень* стека TCP/IP, называемый также *транспортным*.

На этом уровне функционируют протокол управления передачей TCP (Transmission Control Protocol) и протокол дейтаграмм пользователя UDP (User Datagram Protocol). Протокол TCP обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования логических соединений. Этот протокол позволяет равноранговым объектам на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме. TCP позволяет без ошибок доставить сформированный на одном из компьютеров поток байт в любой другой компьютер, входящий в составную сеть. TCP делит поток байт на части - *сегменты*, и передает их ниже лежащему уровню межсетевого взаимодействия. После того как эти сегменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол TCP снова соберет их в непрерывный поток байт.

Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным способом, как и главный протокол уровня межсетевого взаимодействия IP, и выполняет только функции связующего звена (мультиплексора) между сетевым протоколом и многочисленными службами прикладного уровня или пользовательскими процессами.

Прикладной уровень

Прикладной уровень объединяет все службы, предоставляемые системой пользовательским приложениям. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. Прикладной уровень реализуется программными системами, построенными в архитектуре клиент-сервер, базирующимися на протоколах нижних уровней. В отличие от протоколов остальных трех уровней, протоколы прикладного уровня занимаются деталями конкретного приложения и «не интересуются» способами передачи данных по сети. Этот уровень постоянно расширяется за счет присоединения к старым, прошедшим многолетнюю эксплуатацию сетевым службам типа Telnet, FTP, TFTP, DNS, SNMP

сравнительно новых служб таких, например, как протокол передачи гипертекстовой информации HTTP.

Уровень сетевых интерфейсов

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой организации других стеков является интерпретация функций самого нижнего уровня - *уровня сетевых интерфейсов*. Протоколы этого уровня должны обеспечивать интеграцию в составную сеть других сетей, причем задача ставится так: сеть TCP/IP должна иметь средства включения в себя любой другой сети, какую бы внутреннюю технологию передачи данных эта сеть не использовала. Отсюда следует, что этот уровень нельзя определить раз и навсегда. Для каждой технологии, включаемой в составную сеть подсети, должны быть разработаны собственные интерфейсные средства. К таким интерфейсным средствам относятся протоколы инкапсуляции IP-пакетов уровня межсетевого взаимодействия в кадры локальных технологий. Например, документ RFC 1042 определяет способы инкапсуляции IP-пакетов в кадры технологий IEEE 802. Для этих целей должен использоваться заголовок LLC/ SNAP, причем в поле Type заголовка SNAP должен быть указан код 0x0800. Только для протокола Ethernet в RFC 1042 сделано исключение - помимо заголовка LLC/ SNAP разрешается использовать кадр Ethernet DIX, не имеющий заголовка LLC, зато имеющий поле Type. В сетях Ethernet предпочтительным является инкапсуляция IP-пакета в кадр Ethernet DIX.

Уровень сетевых интерфейсов в протоколах TCP/IP не регламентируется, но он поддерживает все популярные стандарты физического и канального уровней: для локальных сетей это Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN, для глобальных сетей - протоколы соединений «точка-точка» SLIP и PPP, протоколы территориальных сетей с коммутацией пакетов X.25, frame relay. Разработана также специальная спецификация, определяющая использование технологии ATM в качестве транспорта канального уровня. Обычно при появлении новой технологии локальных или глобальных сетей она быстро включается в стек TCP/IP за счет разработки соответствующего RFC, определяющего метод инкапсуляции IP-пакетов в ее кадры (спецификация RFC 1577, определяющая работу IP через сети ATM, появилась в 1994 году вскоре после принятия основных стандартов этой технологии).

Соответствие уровней стека TCP/IP семиуровневой модели ISO/OSI

Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем ISO/OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно (рис. 5.6). Рассматривая многоуровневую архитектуру TCP/IP, можно выделить в ней, подобно архитектуре OSI, уровни, функции которых зависят от конкретной технической реализации сети, и уровни, функции которых ориентированны на работу с приложениями (рис. 5.7).

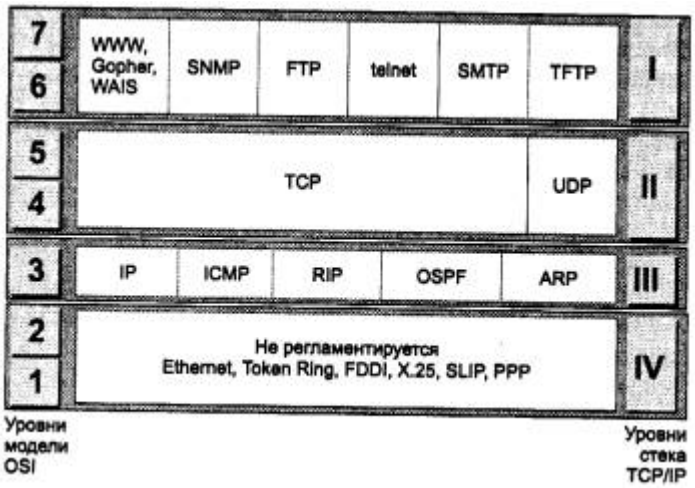


Рис. 5.6. Соответствие уровней стека TCP/IP семиуровневой модели OSI

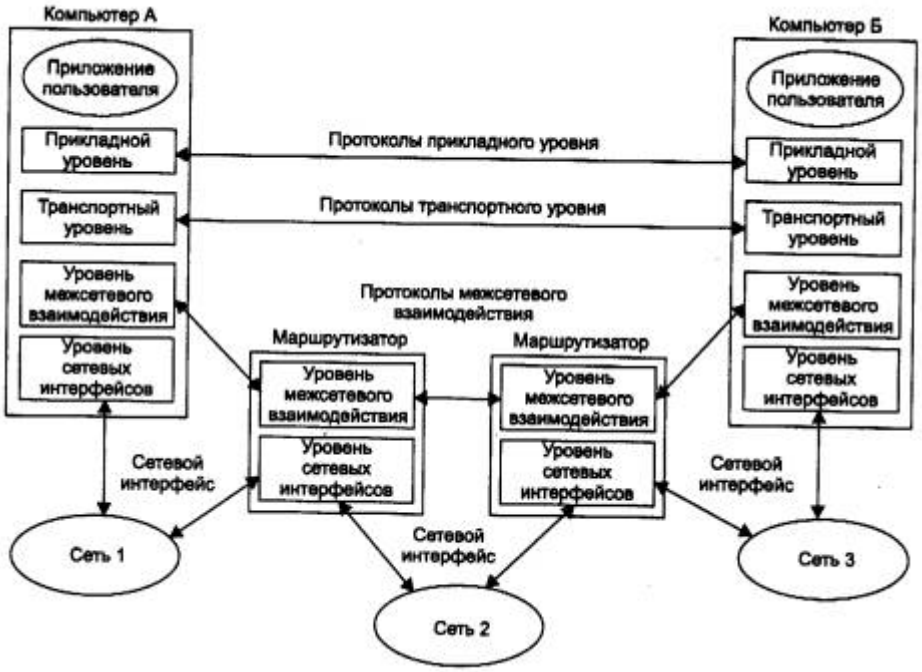


Рис. 5.7. Сетезависимые и сетенезависимые уровни стека TCP/IP

Протоколы прикладного уровня стека TCP/IP работают на компьютерах, выполняющих приложения пользователей. Даже полная смена сетевого оборудования в общем случае не должна влиять на работу приложений, если они получают доступ к сетевым возможностям через протоколы прикладного уровня.

Протоколы транспортного уровня уже более зависят от сети, так как они реализуют интерфейс к уровням, непосредственно организующим передачу данных по сети. Однако, подобно протоколам прикладного уровня, программные модули, реализующие протоколы транспортного уровня, устанавливаются только на конечных узлах. Протоколы двух нижних уровней являются сетезависимыми, а следовательно, программные модули протоколов межсетевого уровня и уровня сетевых интерфейсов устанавливаются как на конечных узлах составной сети, так и на маршрутизаторах.

Каждый коммуникационный протокол оперирует с некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 5.8).

Рис. 5.8. Название единиц данных, используемые в TCP/IP

Потоком называют данные, поступающие от приложений на вход протоколов транспортного уровня TCP и UDP.

Протокол TCP нарезает из потока данных *сегменты*.

Единицу данных протокола UDP часто называют *дейтаграммой* (или датаграммой). Дейтаграмма - это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол межсетевого взаимодействия IP.

Дейтаграмму протокола IP называют также *пакетом*.

В стеке TCP/IP принято называть *кадрами (фреймами)* единицы данных протоколов, на основе которых IP-пакеты переносятся через подсети составной сети. При этом не имеет значения, какое название используется для этой единицы данных в локальной технологии.

Выводы

- Составная сеть (internetwork или internet) - это совокупность нескольких сетей, называемых также подсетями (subnet), которые соединяются между собой маршрутизаторами. Организация совместной транспортной службы в составной сети называется межсетевым взаимодействием (internetworking).
- В функции сетевого уровня входит: передача пакетов между конечными узлами в составных сетях, выбор маршрута, согласование локальных технологий отдельных подсетей.
- Маршрут - это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до пункта назначения. Задачу выбора маршрута из нескольких возможных решают маршрутизаторы и конечные узлы на основе таблиц маршрутизации. Записи в таблицу могут заноситься вручную администратором и автоматически протоколами маршрутизации.
- Протоколы маршрутизации (например, RIP или OSPF) следует отличать от собственно сетевых протоколов (например, IP или IPX). В то время как первые

собирают и передают по сети чисто служебную информацию о возможных маршрутах, вторые предназначены для передачи пользовательских данных.

- Сетевые протоколы и протоколы маршрутизации реализуются в виде программных модулей на конечных узлах-компьютерах и на промежуточных узлах - маршрутизаторах.
- Маршрутизатор представляет собой сложное многофункциональное устройство, в задачи которого входит: построение таблицы маршрутизации, определение на ее основе маршрута, буферизация, фрагментация и фильтрация поступающих пакетов, поддержка сетевых интерфейсов. Функции маршрутизаторов могут выполнять как специализированные устройства, так и универсальные компьютеры с соответствующим программным обеспечением.
- Для алгоритмов маршрутизации характерны одношаговый и многошаговый подходы. Одношаговые алгоритмы делятся на алгоритмы фиксированной, простой и адаптивной маршрутизации. Адаптивные протоколы маршрутизации являются наиболее распространенными и в свою очередь могут быть основаны на дистанционно-векторных алгоритмах и алгоритмах состояния связей.
- Наибольшее распространение для построения составных сетей в последнее время получил стек TCP/IP. Стек TCP/IP имеет 4 уровня: прикладной, основной, уровень межсетевого взаимодействия и уровень сетевых интерфейсов. Соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.
- *Прикладной уровень* объединяет все службы, предоставляемые системой пользовательским приложениям: традиционные сетевые службы типа telnet, FTP, TFTP, DNS, SNMP, а также сравнительно новые, такие, например, как протокол передачи гипертекстовой информации HTTP.
- *На основном уровне* стека TCP/IP, называемом также транспортным, функционируют протоколы TCP и UDP. Протокол управления передачей TCP решает задачу обеспечения надежной информационной связи между двумя конечными узлами. Дейтаграммный протокол UDP используется как экономичное средство связи уровня межсетевого взаимодействия с прикладным уровнем.
- *Уровень межсетевого взаимодействия* реализует концепцию коммутации пакетов в режиме без установления соединений. Основными протоколами этого уровня являются дейтаграммный протокол IP и протоколы маршрутизации (RIP, OSPF, BGP и др.). Вспомогательную роль выполняют протокол межсетевых управляющих сообщений ICMP, протокол группового управления IGMP и протокол разрешения адресов ARP.
- Протоколы *уровня сетевых интерфейсов* обеспечивают интеграцию в составную сеть других сетей. Этот уровень не регламентируется, но поддерживает все популярные стандарты физического и канального уровней: для локальных сетей - Ethernet, Token Ring, FDDI и т. д., для глобальных сетей - X.25, frame relay, PPP, ISDN и т. д.
- В стеке TCP/IP для именования единиц передаваемых данных на разных уровнях используют разные названия: поток, сегмент, дейтаграмма, пакет, кадр.

5.2. Адресация в IP-сетях

5.2.1. Типы адресов стека TCP/IP

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена.

В терминологии TCP/IP под *локальным адресом* понимается такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной интерсети. В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP предполагалось наличие разных типов локальных адресов. Если подсетью интерсети является локальная сеть, то локальный адрес - это MAC - адрес. MAC - адрес назначается сетевым адаптерам и сетевым интерфейсам маршрутизаторов. MAC - адреса назначаются производителями оборудования и являются уникальными, так как управляются централизованно. Для всех существующих технологий локальных сетей MAC - адрес имеет формат 6 байт, например 11-AO-17-3D-BC-01. Однако протокол IP может работать и над протоколами более высокого уровня, например над протоколом IPX или X.25. В этом случае локальными адресами для протокола IP соответственно будут адреса IPX и X.25. Следует учесть, что компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. Некоторые сетевые устройства не имеют локальных адресов. Например, к таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа «точка-точка».

IP-адреса представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. Эти адреса состоят из 4 байт, например 109.26.17.100. IP-адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Internet Network Information Center, InterNIC), если сеть должна работать как составная часть Internet. Обычно поставщики услуг Internet получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами. Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Символьные доменные имена. Символьные имена в IP-сетях называются доменными и строятся по иерархическому признаку. Составляющие полного символьного имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя конечного узла, затем имя группы узлов (например, имя организации), затем имя более крупной группы (поддомена) и так до имени домена самого высокого уровня (например, домена объединяющего организации по географическому принципу: RU - Россия, UK - Великобритания, SU - США), Примеров доменного имени может служить имя base2.sales.zil.ru. Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами,

5.2.2. Классы IP-адресов

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками,

например, 128.10.2.30 - традиционная десятичная форма представления адреса, а 10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая - к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому *классу* относится тот или иной IP-адрес.

На рис. 5.9 показана структура IP-адреса разных классов.

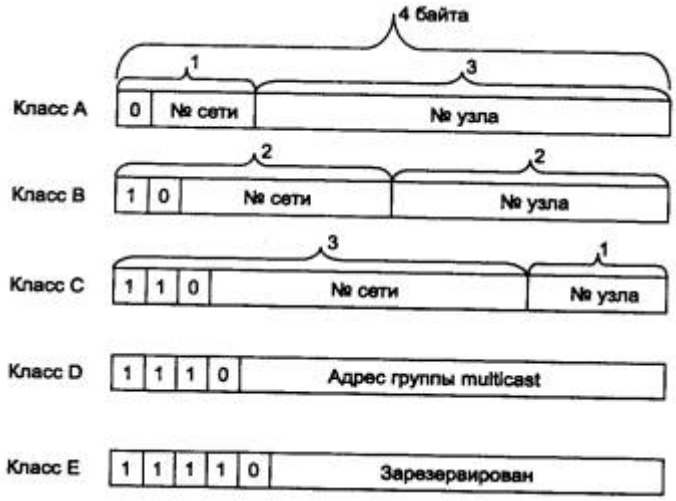


Рис. 5.9. Структура IP-адреса

Если адрес начинается с 0, то сеть относят к *классу A* и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) Сетей класса А немного, зато количество узлов в них может достигать 2^{24}, то есть 16 777 216 узлов.

Если первые два бита адреса равны 10, то сеть относится к *классу B*. В сетях класса В под номер сети и под номер узла отводится по 16 бит, то есть по 2 байта. Таким образом, сеть класса В является сетью средних размеров с максимальным числом узлов 2^{16}, что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть *класса C*. В этом случае под номер сети отводится 24 бита, а под номер узла - 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 28, то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом *класса D и* обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к *классу E,* Адреса этого класса зарезервированы для будущих применений.

В табл. 5.4 приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Таблица 5.4. Характеристики адресов разного класса

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	2^{24}
B	10	128.0.0.0	191.255.0.0	2^{16}
C	110	192.0.1.0	223.255.255.0	2^{8}
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

Большие сети получают адреса класса A, средние - класса B, а маленькие класса C.

5.2.3. Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов.

- Если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP.
- Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет.
- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется *ограниченным широковещательным. сообщением (limited broadcast).*
- Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется *широковещательным сообщением (broadcast).*

При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса C под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот адрес имеет название *loopback*. Можно отнести адрес 127.0.0.0 ко внутренней сети модуля маршрутизации узла, а адрес 127.0.0.1 - к адресу этого модуля на внутренней сети. На самом деле любой адрес сети 127.0.0.0 служит для обозначения своего модуля маршрутизации, а не только 127.0.0.1, например 127.0.0.3.

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам. Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют пределы распространения в интерсети - они ограничены либо сетью, к которой принадлежит узел-источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из составляющих общую сеть частей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Уже упоминавшаяся форма группового IP-адреса - *multicast* - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо группы multicast не обязательно должны принадлежать одной сети. В общем случае они могут распределяться по совершенно различным сетям, находящимся друг от друга на произвольном количестве хопов. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Основное назначение multicast-адресов - распространение информации по схеме «один-ко-многим». Хост, который хочет передавать одну и ту же информацию многим абонентам, с помощью специального протокола IGMP (Internet Group Management Protocol) сообщает о создании в сети новой мультивещательной группы с определенным адресом. Машрутизаторы, поддерживающие мультивещательность, распространяют информацию о создании новой группы в сетях, подключенных к портам этого маршрутизатора. Хосты, которые хотят присоединиться к вновь создаваемой мультивещательной группе, сообщают об этом своим локальным маршрутизаторам и те передают эту информацию хосту, инициатору создания новой группы.

Чтобы маршрутизаторы могли автоматически распространять пакеты с адресом multicast по составной сети, необходимо использовать в конечных маршрутизаторах модифицированные протоколы обмена маршрутной информацией, такие как, например, MOSPF (Multicast OSPF, аналог OSPF).

Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей. Если такие средства найдут широкое применение (сейчас они представляют в основном небольшие экспериментальные островки в общем Internet), то Internet сможет создать серьезную конкуренцию радио и телевидению.

5.2.4. Использование масок в IP-адресации

Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса, который определяется значениями нескольких первых бит адреса. Именно потому, что первый байт адреса 185.23.44.206 попадает в диапазон 128-191, мы можем сказать, что этот адрес относится к классу B, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами - 185.23.0.0, а номером узла - 0.0.44.206.

А что если использовать какой-либо другой признак, с помощью которого можно было ,бы более гибко устанавливать границу между номером сети и номером узла? В качестве такого признака сейчас получили широкое распространение маски. *Маска* - это число,

которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Для стандартных классов сетей маски имеют следующие значения:

- класс А - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс В - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс С-11111111.11111111.11111111.00000000 (255.255.255.0).

ПРИМЕЧАНИЕ Для записи масок используются и другие форматы, например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде: FF.FF.OO.OO - маска для адресов класса В. Часто встречается и такое обозначение 185.23.44.206/16 - эта запись говорит о том, что маска для этого адреса содержит 16 единиц или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, если рассмотренный выше адрес 185.23.44.206 ассоциировать с маской 255.255.255.0, то номером сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде:

IP-адрес 129.64.134.5 - 10000001. 01000000.10000110. 00000101

Маска 255.255.128.0 - 11111111.11111111.10000000. 00000000

Если игнорировать маску, то в соответствии с системой классов адрес 129.64.134.5 относится к классу В, а значит, номером сети являются первые 2 байта - 129.64.0.0, а номером узла - 0.0.134.5.

Если же использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» на IP-адрес, определяют в качестве номера сети в двоичном выражении число:

10000001. 01000000. 10000000. 00000000 или в десятичной форме записи - номер сети 129.64.128.0, а номер узла 0.0.6.5.

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей. На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью

уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов.

5.2.5. Порядок распределения IP-адресов

Номера сетей назначаются либо централизованно, если сеть является частью Internet, либо произвольно, если сеть работает автономно. Номера узлов и в том и в другом случае администратор волен назначать по своему усмотрению, не выходя, разумеется, из разрешенного для этого класса сети диапазона.

Координирующую роль в централизованном распределении IP-адресов до некоторого времени играла организация InterNIC, однако с ростом сети задача распределения адресов стала слишком сложной, и InterNIC делегировала часть своих функций другим организациям и крупным поставщикам услуг Internet.

Уже сравнительно давно наблюдается дефицит IP-адресов. Очень трудно получить адрес класса B и практически невозможно стать обладателем адреса класса A. При этом надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся множество IP-адресов используется нерационально. Очень часто владельцы сети класса C расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве канала связи используют два маршрутизатора, соединенных по схеме «точка-точка» (рис. 5.10). Для вырожденной сети, образованной каналом, связывающим порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети имеются всего 2 узла.

Рис. 5.10. Нерациональное использование пространства IP-адресов

Если же некоторая IP-сеть создана для работы в «автономном режиме», без связи с Internet, тогда администратор этой сети волен назначить ей произвольно выбранный номер. Но и в этой ситуации для того, чтобы избежать каких-либо коллизий, в стандартах Internet определено несколько диапазонов адресов, рекомендуемых для локального использования. Эти адреса не обрабатываются маршрутизаторами Internet ни при каких условиях. Адреса, зарезервированные для локальных целей, выбраны из разных классов; в классе A - это сеть 10.0.0.0, в классе B - это диапазон из 16 номеров сетей 172.16.0.0-172.31.0.0, в классе C - это диапазон из 255 сетей - 192.168.0.0-192.168.255.0.

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию IPv6, в которой резко расширяется адресное пространство за счет использования 16-байтных адресов. Однако и текущая версия IPv4 поддерживает некоторые технологии, направленные на более экономное расходование IP-адресов. Одной из таких технологий является технология *масок* и ее развитие - технология *бесклассовой междоменной маршрутизации (Classless Inker-Domain Routing, CIDR)*. Технология CIDR отказывается от

традиционной концепции разделения адресов протокола IP на классы, что позволяет получать в пользование столько адресов, сколько реально необходимо. Благодаря CIDR поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в точном соответствии с требованиями каждого клиента, при этом у него остается пространство для маневра на случай его будущего роста.

Другая технология, которая может быть использована для снятия дефицита адресов, это *трансляция адресов (Network Address Translator, NAT)*. Узлам внутренней сети адреса назначаются произвольно (естественно, в соответствии с общими правилами, определенными в стандарте), так, как будто эта сеть работает автономно. Внутренняя сеть соединяется с Internet через некоторое промежуточное устройство (маршрутизатор, межсетевой экран). Это промежуточное устройство получает в свое распоряжение некоторое количество внешних «нормальных» IP-адресов, согласованных с поставщиком услуг или другой организацией, распределяющей IP-адреса. Промежуточное устройство способно преобразовывать внутренние адреса во внешние, используя для этого некие таблицы соответствия. Для внешних пользователей все многочисленные узлы внутренней сети выступают под несколькими внешними IP-адресами. При получении внешнего запроса это устройство анализирует его содержимое и при необходимости пересылает его во внутреннюю сеть, заменяя IP-адрес на внутренний адрес этого узла. Процедура трансляции адресов определена в RFC 1631.

5.2.6. Автоматизация процесса назначения IP-адресов

Назначение IP-адресов узлам сети даже при не очень большом размере сети может представлять для администратора утомительную процедуру. Протокол *Dynamic Host Configuration Protocol (DHCP)* освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов.

DHCP может поддерживать способ автоматического динамического распределения адресов, а также более простые способы ручного и автоматического статического назначения адресов. Протокол DHCP работает в соответствии с моделью клиент-сервер. Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP - сервер откликается и посылает сообщение-ответ, содержащее IP-адрес. Предполагается, что DHCP-клиент и DHCP-сервер находятся в одной IP-сети.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, называемое *временем аренды (lease duration)*, что дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Основное преимущество DHCP - автоматизация рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере. Иногда динамическое разделение адресов позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

В ручной процедуре назначения статических адресов активное участие принимает администратор, который предоставляет DHCP - серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному клиенту назначенный администратором адрес.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес из пула наличных IP-адресов без вмешательства оператора. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Адрес дается клиенту из пула в постоянное пользование, то есть с неограниченным сроком аренды. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

DHCP обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие дублирования адресов за счет централизованного управления их распределением. Администратор управляет процессом назначения адресов с помощью параметра «продолжительность аренды», которая определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера в аренду.

Примером работы протокола DHCP может служить ситуация, когда компьютер, являющийся DHCP-клиентом, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей.

DHCP-сервер может назначить клиенту не только IP-адрес клиента, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например, маску, IP-адрес маршрутизатора по умолчанию, IP-адрес сервера DNS, доменное имя компьютера и т. п.

5.2.7. Отображение IP-адресов на локальные адреса

Одной из главных задач, которая ставилась при создании протокола IP, являлось обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии. Непосредственно с решением этой задачи связан уровень межсетевых интерфейсов стека TCP/IP. На этом уровне определяются уже рассмотренные выше спецификации упаковки (инкапсуляции) IP-пакетов в кадры локальных технологий. Кроме этого, уровень межсетевых интерфейсов должен заниматься также крайне важной задачей отображения IP-адресов в локальные адреса.

Для определения локального адреса по IP-адресу используется *протокол разрешения адреса (Address Resolution Protocol, ARP)*. Протокол ARP работает различным образом в зависимости от того, какой протокол канального уровня работает в данной сети - протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещательного доступа одновременно ко всем узлам сети или же протокол глобальной сети (X.25, frame relay), как правило не поддерживающий широковещательный доступ. Существует также протокол, решающий обратную задачу - нахождение IP-адреса по известному локальному адресу. Он называется реверсивным ARP (Reverse Address Resolution Protocol, RARP) и используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

Необходимость в обращении к протоколу ARP возникает каждый раз, когда модуль IP передает пакет на уровень сетевых интерфейсов, например драйверу Ethernet. IP-адрес узла назначения известен модулю IP. Требуется на его основе найти MAC - адрес узла назначения.

Работа протокола ARP начинается с просмотра так называемой *АКР-таблицы* (табл. 5.5). Каждая строка таблицы устанавливает соответствие между IP-адресом и MAC - адресом. Для каждой сети, подключенной к сетевому адаптеру компьютера или к порту маршрутизатора, строится отдельная ARP-таблица.

Таблица 5.5. Пример ARP-таблицы

IP-адрес	MAC - адрес	Тип записи
194.85.135.75	008048EB7E60	Динамический
194.85.135.70	08005A21A722	Динамический
194.85.60.21	008048EB7567	Статический

Поле «Тип записи» может содержать одно из двух значений - «динамический» или «статический». Статические записи создаются вручную с помощью утилиты агр и не имеют срока устаревания, точнее, они существуют до тех пор, пока компьютер или маршрутизатор не будут выключены. Динамические же записи создаются модулем протокола ARP, использующим широковещательные возможности локальных сетевых технологий. Динамические записи должны периодически обновляться. Если запись не обновлялась в течение определенного времени (порядка нескольких минут), то она исключается из таблицы. Таким образом, в ARP - таблице содержатся записи не обо всех узлах сети, а только о тех, которые активно участвуют в сетевых операциях. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют ARP-кэш.

В глобальных сетях администратору сети чаще всего приходится вручную формировать ARP-таблицы, в которых он задает, например, соответствие IP-адреса адресу узла сети X.25, который имеет для протокола IP смысл локального адреса. В последнее время наметилась тенденция автоматизации работы протокола ARP и в глобальных сетях. Для этой цели среди всех маршрутизаторов, подключенных к какой-либо глобальной сети, выделяется специальный маршрутизатор, который ведет ARP-таблицу для всех остальных узлов и маршрутизаторов этой сети. При таком централизованном подходе для всех узлов и маршрутизаторов вручную нужно задать только IP-адрес и локальный адрес выделенного маршрутизатора. Затем каждый узел и маршрутизатор регистрирует свои адреса в выделенном маршрутизаторе, а при необходимости установления соответствия между IP-адресом и локальным адресом узел обращается к выделенному маршрутизатору с запросом и автоматически получает ответ без участия администратора. Работающий таким образом маршрутизатор называют ARP-сервером.

Итак, после того как модуль IP обратился к модулю ARP с запросом на разрешение адреса, происходит поиск в ARP-таблице указанного в запросе IP-адреса. Если таковой адрес в ARP-таблице отсутствует, то исходящий IP-пакет, для которого нужно было определить локальный адрес, ставится в очередь. Далее протокол ARP формирует свой

запрос (ARP-запрос), вкладывает его в кадр протокола канального уровня и рассылает запрос широковещательно.

Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес, а затем отправляет его уже направленно, так как в ARP-запросе отправитель указывает свой локальный адрес. ARP-запросы и ответы используют один и тот же формат пакета. В табл. 5.6 приведены значения полей примера ARP-запроса для передачи по сети Ethernet.

Таблица 5.6. Пример ARP-запроса

Тип сети	1 (0x1)
Тип протокола	2048 (0x800)
Длина локального адреса	6 (0x6)
Длина сетевого адреса	4 (0x4)
Опция	1(0x1)
Локальный адрес отправителя	008048EB7E60
Сетевой адрес отправителя	194.85.135.75
Локальный (искомый) адрес получателя	000000000000
Сетевой адрес получателя	194.85.135.65

В поле «тип сети» для сетей Ethernet указывается значение 1.

Поле «тип протокола» позволяет использовать протокол ARP не только для протокола IP, но и для других сетевых протоколов. Для IP значение этого поля равно 0800 is.

Длина локального адреса для протокола Ethernet равна 6 байт, а длина IP-адреса - 4 байт. В поле операции для ARP-запросов указывается значение 1, если это запрос, и 2, если это ответ.

Из этого запроса видно, что в сети Ethernet узел с IP-адресом 194.85.135.75 пытается определить, какой MAC - адрес имеет другой узел той же сети, сетевой адрес которого 194.85.135.65. Поле искомого локального адреса заполнено нулями.

Ответ присылает узел, опознавший свой IP-адрес. Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет. Протокол IP уничтожает IP-пакеты, направляемые по этому адресу. (Заметим, что протоколы верхнего уровня не могут отличить случай повреждения сети Ethernet от случая отсутствия машины с искомым IP-адресом.) В табл. 5.7 помещены значения полей ARP-ответа, который мог бы поступить на приведенный выше пример ARP-запроса.

Таблица 5.7. Пример ARP-ответа

Тип сети	1 (0x1)
Тип протокола	2048 (0x800)
Длина локального адреса	6 (0x6)
Длина сетевого адреса	4 (0x4)
Опция	1 (0x1)
Локальный адрес отправителя	00E0F77F1920
Сетевой адрес отправителя	194.85.135.65
Локальный (искомый) адрес получателя	008048EB7E60
Сетевой адрес получателя	194.85.135.75

Этот ответ получает машина, сделавшая ARP-запрос. Модуль ARP анализирует ARP-ответ и добавляет запись в свою ARP-таблицу (табл. 5.8). В результате обмена этими двумя ARP-сообщениями модуль IP-узла 194.85.135.75 определил, что IP-адресу 194.85.135.65 соответствует MAC - адрес 00E0F77F1920. Новая запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как она потребовалась.

Таблица 5.8. Обновленная ARP-таблица

IP-адрес	MAC - адрес	Тип записи
194.85.135.75	008048EB7E60	Динамический
194.85.135.70	08005A21A722	Динамический
194.85.60.21	008048EB7567	Статический
194.85.135.65	00E0F77F1920	Динамический

ПРИМЕЧАНИЕ Некоторые реализации IP и ARP не ставят IP-пакеты в очередь на время ожидания ARP-ответов. Вместо этого IP-пакет просто уничтожается, о его восстановление возлагается на модуль TCP или прикладной процесс, работающий через UDP. Такое восстановление выполняется с помощью тайм-аутов и повторных передач. Повторная передача сообщения проходит успешно, так как первая попытка уже вызвала заполнение ARP-таблицы.

5.2.8. Отображение доменных имен на IP-адреса

Организация доменов и доменных имен

Для идентификации компьютеров аппаратное и программное обеспечение в сетях TCP/IP полагается на IP-адреса, поэтому для доступа к сетевому ресурсу в параметрах программы вполне достаточно указать IP-адрес, чтобы программа правильно поняла, к какому хосту ей нужно обратиться. Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным ftp-сервером, а команда http://203.23.106.33 откроет начальную страницу на корпоративном Web-сервере. Однако пользователи обычно предпочитают работать с символьными именами компьютеров, и операционные системы локальных сетей

приучили их к этому удобному способу. Следовательно, в сетях TCP/IP должны существовать символьные имена хостов и механизм для установления соответствия между символьными именами и IP-адресами.

В операционных системах, которые первоначально разрабатывались для работы в локальных сетях, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, то использовались так называемые плоские имена, состоящие из последовательности символов, не разделенных на части. Примерами таких имен являются: NW1_1, mail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC - адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным по нескольким причинам.

Плоские имена не дают возможности разработать единый алгоритм обеспечения уникальности имен в пределах большой сети. В небольших сетях уникальность имен компьютеров обеспечивает администратор сети, записывая несколько десятков имен в журнале или файле. При росте сети задачу решают уже несколько администраторов, согласовывая имена между собой неформальным способом. Однако если сеть расположена в разных городах или странах, то администраторам каждой части сети нужно придумать способ именования, который позволил бы им давать имена новым компьютерам независимо от других администраторов, обеспечивая в то же время уникальность имен для всей сети. Самый надежный способ решения этой задачи - отказ от плоских имен в принципе.

Широковещательный способ установления соответствия между символьными именами и локальными адресами хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где общая широковещательность не поддерживается, нужен другой способ разрешения символьных имен. Обычно хорошей альтернативой широковещательности является применение централизованной службы, поддерживающей соответствие между различными типами адресов всех компьютеров сети. Компания Microsoft для своей корпоративной операционной системы Windows NT разработала централизованную службу WINS, которая поддерживает базу данных NetBIOS-имен и соответствующих им IP-адресов.

Для эффективной организации именования компьютеров в больших сетях естественным является применение иерархических составных имен.

В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей (рис. 5.11).

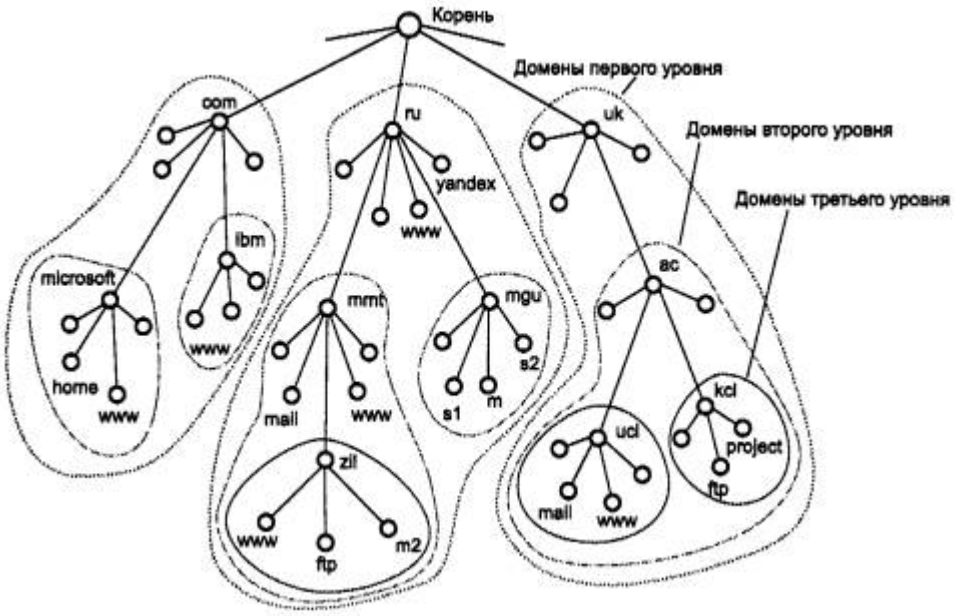


Рис. 5.11. Пространство доменных имен

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяется друг от друга точкой. Например, в имени partnering.microsoft.com составляющая partnering является именем одного из компьютеров в домене Microsoft.com.

Разделение имени на части позволяет *разделить административную ответственность* за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 5.11, один человек может нести ответственность за то, чтобы все имена, которые имеют окончание «та», имели уникальную следующую вниз по иерархии часть. Если этот человек справляется со своими обязанностями, то все имена типа www.ru, mail.mmt.ru или m2.zil.mmt.ru будут отличаться второй по старшинству частью.

Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют *домен* имен *(domain)*. Например, имена wwwl.zil.mmt.ru, ftp.zil.mmt.ru, yandex.ru и sl.mgu.ru входят в домен ru, так как все эти имена имеют одну общую старшую часть - имя ru. Другим примером является домен mgu.ru. Из представленных на рис. 5.11 имен в него входят имена sl.mgu.ru, s2.mgu.ru и rn.mgu.ru. Этот домен образуют имена, у которых две старшие части всегда равны rngu.ru. Имя www.mmt.ru в домен mgu.ru не входит, так как имеет отличающуюся составляющую mmt.

ВНИМАНИЕ Термин «домен» очень многозначен, поэтому его нужно трактовать в рамках определенного контекста. Кроме доменов имен стека TCP/IP в компьютерной литературе также часто упоминаются домены Windows NT, домены коллизий н некоторые другие. Общим у всех этих терминов является то, что они описывают некоторое множество компьютеров, обладающее каким-либо определенным свойством.

Если один домен входит в другой домен как его составная часть, то такой домен могут называть *поддоменом (subdomain)*, хотя название домен за ним также остается. Обычно поддомен называют по имени той его старшей составляющей, которая отличает его от других поддоменов. Например, поддомен mmt.ru обычно называют поддоменом (или доменом) mmt. Имя поддомену назначает администратор вышестоящего домена. Хорошей аналогией домена является каталог файловой системы.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

По аналогии с файловой системой, в доменной системе имен различают краткие имена, относительные имена и полные доменные имена. Краткое имя - это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя - это лист дерева имен. Относительное имя - это составное имя, начинающееся с некоторого уровня иерархии, но не самого верхнего. Например, wwwi.zil - это относительное имя. *Полное доменное имя (fully qualified domain name, FQJDN)* включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: wwwl.zil.mmt.ru.

Необходимо подчеркнуть, что компьютеры входят в домен в соответствии со своими составными именами, при этом они могут иметь совершенно различные IP-адреса, принадлежащие к различным сетям и подсетям. Например, в домен mgu.ru могут входить хосты с адресами 132.13.34.15, 201.22.100.33,14.0.0.6. Доменная система имен реализована в сети Internet, но она может работать и как автономная система имен в крупной корпоративной сети, использующей стек TCP/IP, но не связанной с Internet.

В Internet корневой домен управляется центром InterNIC. Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, а для различных типов организаций - следующие обозначения:

- corn - коммерческие организации (например, microsoft.com);
- edu - образовательные (например, mitedu);
- gov - правительственные организации (например, nsf.gov);
- org - некоммерческие организации (например, fidonet.org);
- net - организации, поддерживающие сети (например, nsf.net).

Каждый домен администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой InterNIC делегировал свои полномочия по

распределению имен доменов. В России такой организацией является РосНИИРОС, которая отвечает за делегирование имен поддоменов в домене ru.

Система доменных имен DNS

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста, так и средствами централизованной службы. На раннем этапе развития Internet на каждом хосте вручную создавался текстовый файл с известным именем hosts. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «IP-адрес - доменное имя», например 102.54.94.97 - rhino.acme.com.

По мере роста Internet файлы hosts также росли, и создание масштабируемого решения для разрешения имен стало необходимостью.

Таким решением стала специальная служба - *система доменных имен (Domain Name System, DNS)*. DNS - это централизованная служба, основанная на распределенной базе отображений «доменное имя - IP-адрес». Служба DNS использует в своей работе протокол типа «клиент-сервер». В нем определены DNS-серверы и DNS-кли-енты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиен-ты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес.

Служба DNS использует текстовые файлы почти такого формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый сервер службы DNS хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. Этот сервер может хранить отображения «доменное имя - IP-адрес» для всего домена, включая все его поддомены. Однако при этом решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. (Аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих».) Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более-менее равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmtru будет хранить отображения для всех имен, заканчивающихся на mmt.ru: wwwl.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д. Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер кроме таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых являются широко известными (их можно узнать, например, в InterNIC).

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Действительно, в обоих случаях

составное имя отражает иерархическую структуру организации соответствующих справочников - каталогов файлов или таблиц DNS. Здесь домен и доменный DNS-сервер являются аналогом каталога файловой системы. Для доменных имен, так же как и для символьных имен файлов, характерна независимость именования от физического местоположения.

Процедура поиска адреса файла по символьному имени заключается в последовательном просмотре каталогов, начиная с корневого. При этом предварительно проверяется кэш и текущий каталог. Для определения IP-адреса по доменному имени также необходимо просмотреть все DNS-серверы, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена. Существенным же отличием является то, что файловая система расположена на одном компьютере, а служба DNS по своей природе является распределенной.

Существуют две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

- DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени;
- DNS-сервер отвечает, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени;
- DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая схема взаимодействия называется нерекурсивной или итеративной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Так как эта схема загружает клиента достаточно сложной работой, то она применяется редко.

Во втором варианте реализуется рекурсивная процедура:

- DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, который обслуживает поддомен, к которому принадлежит имя клиента;
- если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту; это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также может соответствовать случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше;
- если же локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в первом варианте; получив ответ, он передает его клиенту, который все это время просто ждал его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется косвенной или рекурсивной. Практически все DNS-клиенты используют рекурсивную процедуру.

Для ускорения поиска IP-адресов DNS-серверы широко применяют процедуру кэширования проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на определенное время - обычно от нескольких часов до нескольких дней.

Выводы

- В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена. Все эти типы адресов присваиваются узлам составной сети независимо друг от друга.
- IP-адрес имеет длину 4 байта и состоит из номера сети и номера узла. Для определения границы, отделяющей номер сети от номера узла, реализуются два подхода. Первый основан на понятии класса адреса, второй - на использовании масок.
- Класс адреса определяется значениями нескольких первых бит адреса. В адресах класса A под номер сети отводится один байт, а остальные три байта - под номер узла, поэтому они используются в самых больших сетях. Для небольших сетей больше подходят адреса класса **C**, в которых номер сети занимает три байта, а для нумерации узлов может быть использован только один байт. Промежуточное положение занимают адреса класса B.
- Другой способ определения, какая часть адреса является номером сети, а какая номером узла, основан на использовании маски. Маска - это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые в IP-адресе должны интерпретироваться как номер сети.
- Номера сетей назначаются либо централизованно, если сеть является частью Internet, либо произвольно, если сеть работает автономно.
- Процесс распределения IP-адресов по узлам сети может быть автоматизирован с помощью протокола DHCP.
- Установление соответствия между IP-адресом и аппаратным адресом (чаще всего MAC - адресом) осуществляется протоколом разрешения адресов ARP, который для этой цели просматривает ARP-таблицы. Если нужный адрес отсутствует, то выполняется широковещательный ARP-запрос.
- В стеке TCP/IP применяется доменная система символьных имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей. Совокупность имен, у которых несколько старших составных частей совпадают, образуют домен имен. Доменные имена назначаются централизованно, если сеть является частью Internet, в противном случае - локально.
- Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста с использованием файла hosts, так и с помощью централизованной службы DNS, основанной на распределенной базе отображений «доменное имя - IP-адрес».

5.3. Протокол IP

5.3.1. Основные функции протокола IP

Основу транспортных средств стека протоколов TCP/IP составляет *протокол межсетевого взаимодействия (Internet Protocol, IP)*. Он обеспечивает передачу дейтаграмм от отправителя к получателям через объединенную систему компьютерных сетей.

Название данного протокола - Intrenet Protocol - отражает его суть: он должен передавать пакеты *между сетями*. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их

помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель.

Протокол IP относится к протоколам без установления соединений. Перед IP не ставится задача надежной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как независимую единицу, не имеющую связи ни с какими другими IP-пакетами. В протоколе IP нет механизмов, обычно применяемых для увеличения достоверности конечных данных: отсутствует квитирование - обмен подтверждениями между отправителем и получателем, нет процедуры упорядочивания, повторных передач или других подобных функций. Если во время продвижения пакета произошла какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен по причине истечения времени жизни или из-за ошибки в контрольной сумме, то модуль IP не пытается заново послать испорченный или потерянный пакет. Все вопросы обеспечения надежности доставки данных по составной сети в стеке TCP/IP решает протокол TCP, работающий непосредственно над протоколом IP. Именно TCP организует повторную передачу пакетов, когда в этом возникает необходимость.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов (например, от сетевого протокола IPX), является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными, максимально допустимыми значениями поля данных кадров MTU. Свойство фрагментации во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

Имеется прямая связь между функциональной сложностью протокола и сложностью заголовка пакетов, которые этот протокол использует. Это объясняется тем, что основные служебные данные, на основании которых протокол выполняет то или иное действие, переносятся между двумя модулями, реализующими этот протокол на разных машинах, именно в полях заголовков пакетов. Поэтому очень полезно изучить назначение каждого поля заголовка IP-пакета, и это изучение дает не только формальные знания о структуре пакета, но и объясняет все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

5.3.2. Структура IP-пакета

IP-пакет состоит из заголовка и поля данных. Заголовок, как правило, имеющий длину 20 байт, имеет следующую структуру (рис. 5.12).

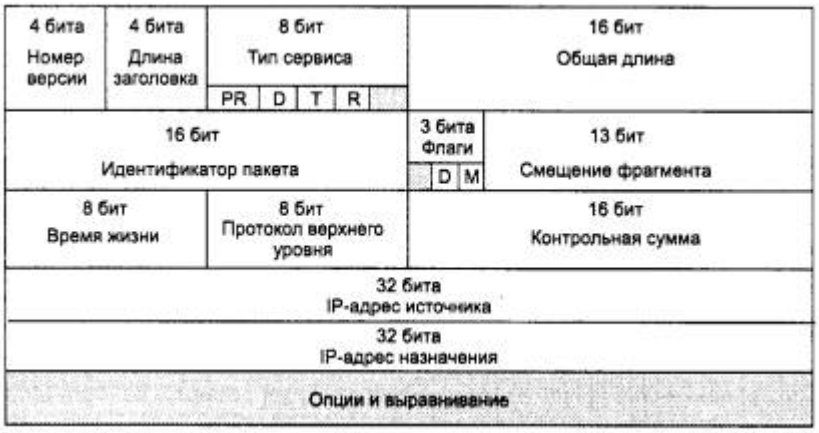


Рис. 5.12. Структура заголовка IP-пакета

Поле *Номер версии (Version),* занимающее 4 бит, указывает версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4), и готовится переход на версию 6 (IPv6).

Поле *Длина заголовка (IHL)* IP-пакета занимает 4 бит и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байт в поле *Опции (IP Options).* Наибольший заголовок занимает 60 октетов.

Поле *Тип сервиса (Type of Service)* занимает один байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе *приоритета* пакета *(Precedence),* Приоритет может иметь значения от самого низкого - 0 (нормальный пакет) до самого высокого - 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Поле *Тип сервиса* содержит также три бита, определяющие критерий выбора маршрута. Реально выбор осуществляется между тремя альтернативами: малой задержкой, высокой достоверностью и высокой пропускной способностью. Установленный бит D (delay) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T - для максимизации пропускной способности, а бит R - для максимизации надежности доставки. Во многих сетях улучшение одного из этих параметров связано с ухудшением другого, кроме того, обработка каждого из них требует дополнительных вычислительных затрат. Поэтому редко, когда имеет смысл устанавливать одновременно хотя бы два из этих трех критериев выбора маршрута. Зарезервированные биты имеют нулевое значение.

Поле *Общая длина (Total Length)* занимает 2 байта и означает общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт, однако в большинстве хост-компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной в 1500 байт, умещающиеся в поле данных кадра Ethernet. В стандарте предусматривается, что все хосты должны быть готовы принимать пакеты вплоть до 576 байт длиной (приходят ли они целиком или по фрагментам). Хостам рекомендуется отправлять пакеты размером более чем 576 байт, только если они уверены,

что принимающий хост или промежуточная сеть готовы обслуживать пакеты такого размера.

Поле *Идентификатор пакета (Identification)* занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля.

Поле *Флаги (Flags)* занимает 3 бита и содержит признаки, связанные с фрагментацией. Установленный бит DF (Do not Fragment) запрещает маршрутизатору фрагментировать данный пакет, а установленный бит MF (More Fragments) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле *Смещение фрагмента (Fragment Offset)* занимает 13 бит и задает смещение в байтах поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами MTU. Смещение должно быть кратно 8 байт.

Поле *Время жизни (Time to Live)* занимает один байт и означает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеряется в секундах и задается источником передачи. На маршрутизаторах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица; единица вычитается и в том случае, когда время задержки меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за одну секунду, то время жизни можно считать равным максимальному числу узлов, которые разрешено пройти данному пакету до того, как он достигнет места назначения. Если параметр времени жизни станет нулевым до того, как пакет достигнет получателя, этот пакет будет уничтожен. Время жизни можно рассматривать как часовой механизм самоуничтожения. Значение этого поля изменяется при обработке заголовка IP-пакета.

Идентификатор *Протокол верхнего уровня (Protocol)* занимает один байт и указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета (например, это могут быть сегменты протокола TCP, дейтаграммы UDP, пакеты ICMP или OSPF). Значения идентификаторов для различных протоколов приводятся в документе RFC «Assigned Numbers».

Контрольная сумма (Header Checksum) занимает 2 байта и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, время жизни), контрольная сумма проверяется и повторно рассчитывается при каждой обработке IP-заголовка. Контрольная сумма - 16 бит - подсчитывается как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля «контрольная сумма» устанавливается в нуль. Если контрольная сумма неверна, то пакет будет отброшен, как только ошибка будет обнаружена.

Поля *IP-адрес источника (Source IP Address)* и *IP-адрес назначения (Destination IP Address)* имеют одинаковую длину - 32 бита - и одинаковую структуру.

Поле *Опции (IP Options)* является необязательным и используется обычно только при отладке сети. Механизм опций предоставляет функции управления, которые необходимы или просто полезны при определенных ситуациях, однако он не нужен при обычных коммуникациях. Это поле состоит из нескольких подполей, каждое из которых может

быть одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки. Так как число подполей может быть произвольным, то в конце поля *Опции* должно быть добавлено несколько байт для выравнивания заголовка пакета по 32-битной границе.

Поле *Выравнивание (Padding)* используется для того, чтобы убедиться в том, что IP-заголовок заканчивается на 32-битной границе. Выравнивание осуществляется нулями.

Ниже приведена распечатка значений полей заголовка одного из реальных IP-пакетов, захваченных в сети Ethernet средствами анализатора протоколов Microsoft Network Monitor.

IP Version = 4 (0x4)

IP Header Length = 20 (0x14)

IP Service Type = 0 (0x0)

IP Precedence = Routine

IP ...0.... = Normal Delay

IP ....0... = Normal Throughput

IP .....0.. = Normal Reliability

IP Total Length = 54 (0x36)

IP Identification = 31746 (0x7C02)

IP Flags Summary ° 2 (0x2)

IP ..... 0 = Last fragment in datagram

IP ..... 1. = Cannot fragment datagram

IP Fragment Offset = 0 (0x0) bytes

IP Time to Live = 128 (0x80)

IP Protocol = TCP - Transmission Control

IP Checksum =0xEB86

IP Source Address = 194.85.135.75

IP Destination Address = 194.85.135.66

IP Data: Number of data bytes remaining = 34 (0x0022)

5.3.3. Таблицы маршрутизации в IP-сетях

Программные модули протокола IP устанавливаются на всех конечных станциях и маршрутизаторах сети. Для продвижения пакетов они используют таблицы маршрутизации.

Примеры таблиц различных типов маршрутизаторов

Структура таблицы маршрутизации стека TCP/IP соответствует общим принципам построения таблиц маршрутизации, рассмотренным выше. Однако важно отметить, что вид таблицы IP-маршрутизации зависит от конкретной реализации стека TCP/IP. Приведем пример трех вариантов таблицы маршрутизации, с которыми мог бы работать маршрутизатор Ml в сети, представленной на рис. 5.13.

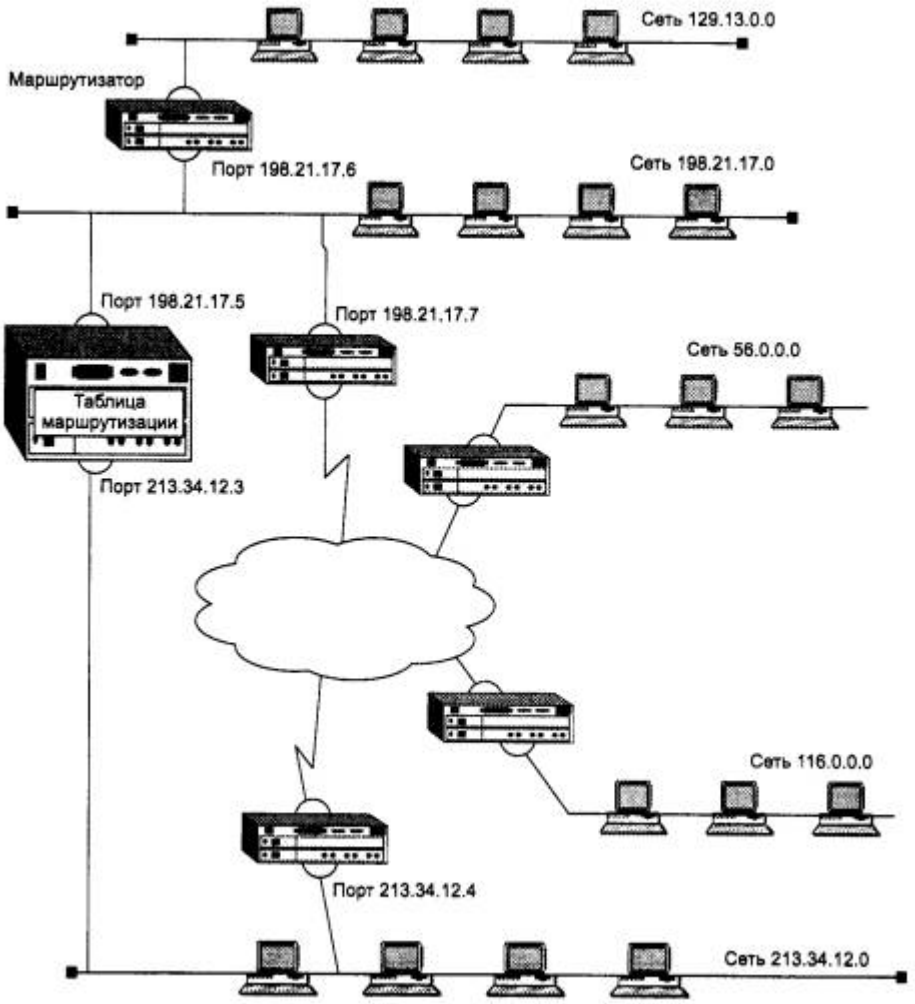


Рис. 5.13. Пример маршрутизируемой сети

Если представить, что в качестве маршрутизатора Ml в данной сети работает штатный программный маршрутизатор MPR операционной системы Microsoft Windows NT, то его таблица маршрутизации могла бы иметь следующий вид (табл. 5.9).

Таблица 5.9. Таблица программного маршрутизатора MPR Windows NT

Network Address	Netmask	Gateway Address	Interface	Metric
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0	1
0.0.0.0	0.0.0.0	198.21.17.7	198.21.17.5	1
56.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	255.255.0.0	198.21.17.6	198.21.17.5	2
198.21.17.0	255.255.255.0	198.21.17.5	198.21.17.5	1
198.21.17.5	255.255.255.255	127.0.0.1	127.0.0.1	1
198.21.17.255	255.255.255.255	198.21.17.5	198.21.17.5	1
213.34.12.0	255.255.255.0	213.34.12.3	213.34.12.3	1
213.34.12.3	255.255.255.255	127.0.0.1	127.0.0.1	1
213.34.12.255	255.255.255.255	213.34.12.3	213.34.12.3	1
224.0.0.0	224.0.0.0	198.21.17.6	198.21.17.6	1
224.0.0.0	224.0.0.0	213.34.12.3	213.34.12.3	1
255.255.255.255	255.255.255.255	198.21.17.6	198.21.17.6	1

Если на месте маршрутизатора M1 установить аппаратный маршрутизатор NetBuilder II компании 3 Com, то его таблица маршрутизации для этой же сети может выглядеть так, как показано в табл. 5.10.

Таблица 5.10. Таблица маршрутизации аппаратного маршрутизатора NetBuilder II компании 3 Com

NetBuilder# Show — IP AllRoutes
Total Routes = 5 Total Direct Networks = 2

Destination	Mask	Gateway	Metric	Status	TTL	Source
198.21.17.0	255.255.255.0	198.21.17.5	0	Up	—	Connected
213.34.12.0	255.255.255.0	213.34.12.3	0	Up	—	Connected
56.0.0.0	255.0.0.0	213.34.12.4	14	Up	—	Static
116.0.0.0	255.0.0.0	213.34.12.4	12	Up	—	Static
129.13.0.0	255.255.0.0	198.21.17.6	1	Up	160	RIP

Таблица 5.11 представляет собой таблицу маршрутизации для маршрутизатора Ml, реализованного в виде программного маршрутизатора одной из версий операционной системы Unix.

Таблица 5.11. Таблица маршрутизации Unix-маршрутизатора

Destination	Gateway	Flags	Refcnt	Use	Interface
127.0.0.0	127.0.01	UH	1	154	lo0
Default	198.21.17.7	UG	5	43270	le0
198.21.17.0	198.21.17.5	U	35	246876	le0
213.34.12.0	213.34.12.3	U	44	132435	le1
129.13.0.0	198.21.1.7.6	UG	6	16450	le0
56.0.0.0	213.34.12.4	UG	12	5764	le1
116.0.0.0	213.34.12.4	UG	21	23544	le1

ПРИМЕЧАНИЕ Заметим, что поскольку между структурой сети и таблицей маршрутизации в принципе нет однозначного соответствия, то и для каждого из приведенных вариантов таблицы можно предложить свои «подварианты», отличающиеся выбранным маршрутом к той или иной сети. В данном случае внимание концентрируется на существенных различиях в форме представления маршрутной информации разными реализациями маршрутизаторов.

Назначение полей таблицы маршрутизации

Несмотря на достаточно заметные внешние различия, во всех трех таблицах есть все те ключевые параметры, необходимые для работы маршрутизатора, которые были рассмотрены ранее при обсуждении концепции маршрутизации.

К таким параметрам, безусловно, относятся адрес сети назначения (столбцы «Destination» в маршрутизаторах NetBuilder и Unix или «Network Address» в маршрутизаторе MPR) и адрес следующего маршрутизатора (столбцы «Gateway» в маршрутизаторах NetBuilder и Unix или «Gateway Address» в маршрутизаторе MPR).

Третий ключевой параметр - адрес порта, на который нужно направить пакет, в некоторых таблицах указывается прямо (поле «Interface» в таблице Windows NT), а в некоторых - косвенно. Так, в таблице Unix-маршрутизатора вместо адреса порта задается его условное наименование - 1e0 для порта с адресом 198.21.17.5, lei для порта с адресом 213.34.12.3 и 1o0 для внутреннего порта с адресом 127.0.0.1.

В маршрутизаторе NetBuilder II поле, обозначающее выходной порт в какой-либо форме, вообще отсутствует. Это объясняется тем, что адрес выходного порта всегда можно косвенно определить по адресу следующего маршрутизатора. Например, попробуем определить по табл. 5.10 адрес выходного порта для сети 56.0.0.0. Из таблицы следует, что следующим маршрутизатором для этой сети будет маршрутизатор с адресом 213.34.12.4. Адрес следующего маршрутизатора должен принадлежать одной из непосредственно присоединенных к маршрутизатору сетей, и в данном случае это сеть 213.34.12.0. Маршрутизатор имеет порт, присоединенный к этой сети, и адрес этого порта 213.34.12.3 мы находим в поле «Gateway» второй строки таблицы маршрутизации, которая описывает непосредственно присоединенную сеть 213.34.12.0. Для непосредственно присоединенных сетей адресом следующего маршрутизатора всегда является адрес собственного порта маршрутизатора. Таким образом, адрес выходного порта для сети 56.0.0 - это адрес 213.34.12.3.

Остальные параметры, которые можно найти в представленных версиях таблицы маршрутизации, являются необязательными для принятия решения о пути следования пакета.

Наличие или отсутствие поля маски в таблице говорит о том, насколько современен данный маршрутизатор. Стандартным решением сегодня является использование поля маски в каждой записи таблицы, как это сделано в таблицах маршрутизаторов MPR Windows NT (поле «Netmask») и NetBuilder (поле «Mask»). Обработка масок при принятии решения маршрутизаторами будет рассмотрена ниже. Отсутствие поля маски говорит о

том, что либо маршрутизатор рассчитан на работу только с тремя стандартными классами адресов, либо он использует для всех записей одну и ту же маску, что снижает гибкость маршрутизации.

Метрика, как видно из примера таблицы Unix-маршрутизатора, является необязательным параметром. В остальных двух таблицах это поле имеется, однако оно используется только в качестве признака непосредственно подключенной сети. Действительно, если в таблице маршрутизации каждая сеть назначения упомянута только один раз, то поле метрики не будет приниматься во внимание при выборе маршрута, так как выбор отсутствует. А вот признак непосредственно подключенной сети маршрутизатору нужен, поскольку пакет для этой сети обрабатывается особым способом - он не передается следующему маршрутизатору, а отправляется узлу назначения. Поэтому метрика 0 для маршрутизатора NetBuilder или 1 для маршрутизатора MPR просто говорит маршрутизатору, что эта сеть непосредственно подключена к его порту, а другое значение метрики соответствует удаленной сети. Выбор значения метрики для непосредственно подключенной сети является достаточно произвольным, главное, чтобы метрика удаленной сети отсчитывалась с учетом этого выбранного начального значения. В Unix-маршрутизаторе используется поле признаков, где флаг G отмечает удаленную сеть, а его отсутствие - непосредственно подключенную.

Однако существуют ситуации, когда маршрутизатор должен обязательно хранить значение метрики для записи о каждой удаленной сети. Эти ситуации возникают, когда записи в таблице маршрутизации являются результатом работы некоторых протоколов маршрутизации, например протокола RIP. В таких протоколах новая информация о какой-либо удаленной сети сравнивается с имеющейся в таблице, и если метрика новой информации лучше имеющейся, то новая запись вытесняет имеющуюся. В таблице Unix-маршрутизатора поле метрики отсутствует, и это значит, что он не использует протокол RIP.

Флаги записей присутствуют только в таблице Unix-маршрутизатора. Они описывают характеристики записи.

- U - показывает, что маршрут активен и работоспособен. Аналогичный смысл имеет поле «Status» в маршрутизаторе NetBuilder.
- H - признак специфического маршрута к определенному хосту. Маршрут ко всей сети, к которой принадлежит данный хост, может отличаться от данного маршрута.
- G - означает, что маршрут пакета проходит через промежуточный маршрутизатор (gateway). Отсутствие этого флага отмечает непосредственно подключенную сеть.
- D - означает, что маршрут получен из сообщения Redirect (перенаправление) протокола ICMP. Этот признак может присутствовать только в таблице маршрутизации конечного узла. Признак означает, что конечный узел в какой-то предыдущей передаче пакета выбрал не самый рациональный следующий маршрутизатор на пути к данной сети, и этот маршрутизатор с помощью протокола ICMP сообщил, что все последующие пакеты к данной сети нужно отправлять через другой следующий маршрутизатор. Протокол ICMP может посылать сообщения только узлу-отправителю, поэтому у промежуточного маршрутизатора этот признак встретиться не может. Признак никак не влияет на процесс маршрутизации, он только указывает администратору источник появления записи. В таблице Unix-маршрутизатора используются еще два поля, имеющих справочное значение. Поле «Refcnt» показывает, сколько раз на данный маршрут ссылались

при продвижении пакетов. Поле «Use» отражает количество пакетов, переданных по данному маршруту.

В таблице маршрутизатора NetBuilder также имеются два справочных поля. Поле времени жизни «TTL» (Time To Live) имеет смысл для динамических записей, которые имеют ограниченный срок жизни. Текущее значение поля показывает оставшийся срок жизни записи в секундах. Поле «Source» отражает источник появления записи в таблице маршрутизации. Хотя это поле имеется не во всех маршрутизаторах, но практически для всех маршрутизаторов существуют три основных источника появления записи в таблице.

Источники и типы записей в таблице маршрутизации

Первым источником является программное обеспечение стека TCP/IP. При инициализации маршрутизатора это программное обеспечение автоматически заносит в таблицу несколько записей, в результате чего создается так называемая *минимальная таблица маршрутизации*.

Это, во-первых, записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию, информация о которых появляется в стеке при ручном конфигурировании интерфейсов компьютера или маршрутизатора. К таким записям в приведенных примерах относятся записи о сетях 213.34.12.0 и 198.21.17.0, а также запись о маршрутизаторе по умолчанию - default в Unix-маршрутизаторе и 0.0.0.0 в маршрутизаторе MPR Windows NT, В приведенном примере таблицы для маршрутизатора NetBuilder маршрутизатор по умолчанию не используется, следовательно, при поступлении пакета с адресом назначения, отсутствующим в таблице маршрутизации, этот пакет будет отброшен.

Во-вторых, программное обеспечение автоматически заносит в таблицу маршрутизации записи об адресах особого назначения. В приведенных примерах таблица маршрутизатора MPR Windows NT содержит наиболее полный набор записей такого рода. Несколько записей в этой таблице связаны с особым адресом 127.0.0.0 (loopback), который используется для локального тестирования стека TCP/IP. Пакеты, направленные в сеть с номером 127.0.0.0, не передаются протоколом IP на канальный уровень для последующей передачи в сеть, а возвращаются в источник - локальный модуль IP. Записи с адресом 224.0.0.0 требуются для обработки групповых адресов (multicast address). Кроме того, в таблицу могут быть занесены адреса, предназначенные для обработки широковещательных рассылок (например, записи 8 и 11 содержат адрес отправки широковещательного сообщения в соответствующих подсетях, а последняя запись в таблице - адрес ограниченной широковещательной рассылки сообщения). Заметим, что в некоторых таблицах записи об особых адресах вообще отсутствуют.

Вторым источником появления записи в таблице является администратор, непосредственно формирующий запись с помощью некоторой системной утилиты, например программы route, имеющейся в операционных системах Unix и Windows NT. В аппаратных маршрутизаторах также всегда имеется команда для ручного задания записей таблицы маршрутизации. Заданные вручную записи всегда являются статическими, то есть не имеют срока истечения жизни. Эти записи могут быть как постоянными, то есть сохраняющимися при перезагрузке маршрутизатора, так и временными, хранящимися в таблице только до выключения устройства. Часто администратор вручную заносит запись default о маршрутизаторе по умолчанию. Таким же образом в таблицу маршрутизации может быть внесена запись о специфичном для узла маршруте. Специфичный для узла маршрут содержит вместо номера сети полный IP-адрес, то есть адрес, имеющий

ненулевую информацию не только в поле номера сети, но и в поле номера узла. Предполагается, что для такого конечного узла маршрут должен выбираться не так, как для всех остальных узлов сети, к которой он относится. В случае когда в таблице есть разные записи о продвижении пакетов для всей сети и ее отдельного узла, при поступлении пакета, адресованного узлу, маршрутизатор отдаст предпочтение записи с полным адресом узла.

И наконец, третьим источником записей могут быть протоколы маршрутизации, такие как RIP или OSPF. Такие записи всегда являются динамическими, то есть имеют ограниченный срок жизни. Программные маршрутизаторы Windows NT и Unix не показывают источник появления той или иной записи в таблице, а маршрутизатор NetBuilder использует для этой цели поле «Source». В приведенном в табл. 5.10 примере первые две записи созданы программным обеспечением стека на основании данных о конфигурации портов маршрутизатора - это показывает признак «Connected». Следующие две записи обозначены как «Static», что указывает на то, что их ввел вручную администратор. Последняя запись является следствием работы протокола RIP, поэтому в ее поле «TTL» имеется значение 160.

5.3.4. Маршрутизация без использования масок

Рассмотрим на примере IP-сети (рис. 5.14) алгоритм работы средств сетевого уровня по продвижению пакета в составной сети. При этом будем считать, что все узлы сети, рассматриваемой в примере, имеют адреса, основанные на классах, без использования масок. Особое внимание будет уделено взаимодействию протокола IP с протоколами разрешения адресов ARP и DNS.

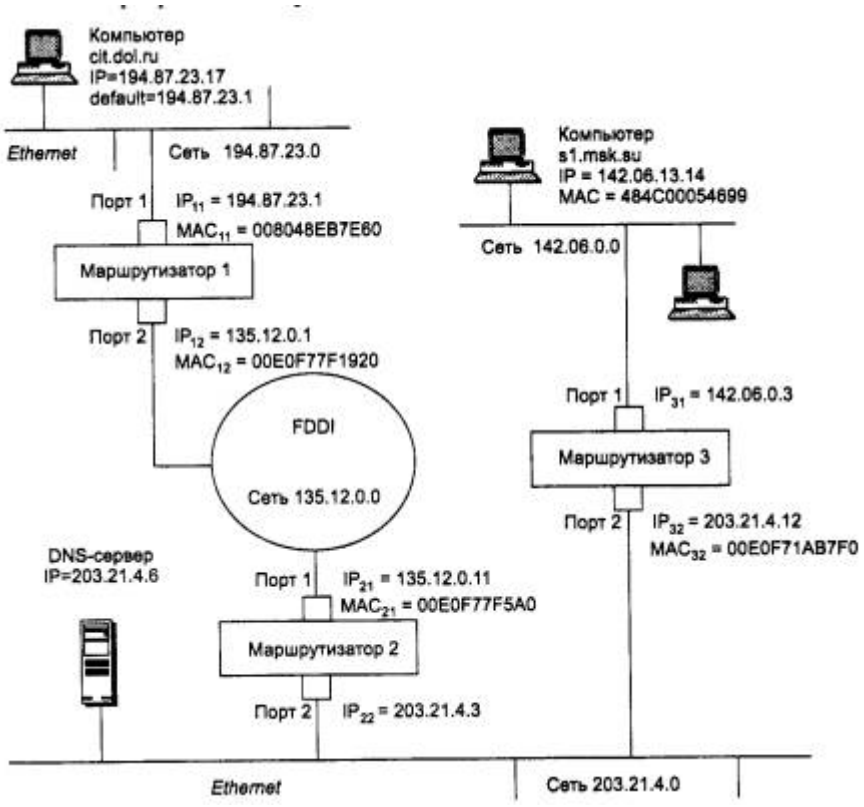


Рис. 5.14. Пример взаимодействия компьютеров через сеть

1. Итак, пусть пользователь компьютера cit.dol.ru, находящегося в сети Ethernet и имеющего IP-адрес 194.87.23.17 (адрес класса C), обращается по протоколу FTP к компьютеру sl.msk.su, принадлежащему другой сети Ethernet и имеющему IP-адрес 142.06,13.14 (адрес класса B): > ftp sl.msk.su

   Модуль FTP упаковывает свое сообщение в сегмент транспортного протокола TCP, который в свою очередь помещает свой сегмент в пакет протокола IP. В заголовке IP-пакета должен быть указан IP-адрес узла назначения. Так как пользователь компьютера cit.dol.ru использует символьное имя компьютера sl.msk.su, то стек TCP/IP должен определить IP-адрес узла назначения самостоятельно.

   При конфигурировании стека TCP/IP в компьютере cit.dol.ru был задан его собственный IP-адрес, IP-адрес маршрутизатора по умолчанию и IP-адрес DNS-сервера. Модуль IP может сделать запрос к серверу DNS, но обычно сначала просматривается локальная таблица соответствия символьных имен и IP-адресов. Такая таблица хранится чаще всего в виде текстового файла простой структуры - каждая его строка содержит запись об одном символьном имени и его IP-адресе. В ОС Unix такой файл традиционно носит имя hosts и находится в каталоге /etc.

2. Будем считать, что компьютер dt.dol.ru имеет файл hosts, а в нем есть строка 142.06.13.14 sl.msk.su.

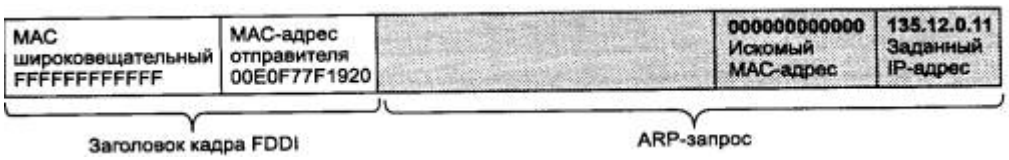
   Таким образом, разрешение имени выполняется локально, и протокол IP может теперь формировать IP-пакеты с адресом назначения 142.06.13.14 для взаимодействия с компьютером sl.msk.su.

3. Модуль IP компьютера cit.dol.ru проверяет, нужно ли маршрутизировать пакеты с адресом 142.06.13.14. Так как адрес сети назначения (142.06.0.0) не совпадает с адресом (194.87.23.0) сети, которой принадлежит компьютер-отправитель, то маршрутизация необходима.
4. Компьютер cit.dol.ru начинает формировать кадр Ethernet для отправки IP-пакета маршрутизатору по умолчанию, IP-адрес которого известен - 194.87.23.1, но неизвестен MAC - адрес, необходимый для перемещения кадра в локальной сети. Для определения MAC - адреса маршрутизатора протокол IP обращается к протоколу ARP, который просматривает ARP-таблицу. Если в последнее время компьютер cit.dol.ru выполнял какие-либо межсетевые обмены, то скорее всего искомая запись, содержащая соответствие между IP- и MAC - адресами маршрутизатора по умолчанию уже находится в кэш-таблице протокола ARP. Пусть в данном случае нужная запись была найдена именно в кэш-таблице: 194.87.23.1 008048EB7E60

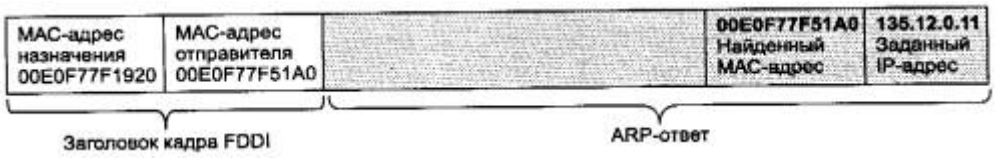
   Обозначим найденный MAC - адрес 008048EB7E60 в соответствии с номером маршрутизатора и его порта через MAC_{11}.

5. В результате компьютер cit.dol.ru отправляет по локальной сети пакет, упакованный в кадр Ethernet, имеющий следующие поля:

6. Кадр принимается портом 1 маршрутизатора 1 в соответствии с протоколом Ethernet, так как MAC - узел этого порта распознает свой адрес MAC_{11}. Протокол Ethernet извлекает из этого кадра IP-пакет и передает его программному обеспечению маршрутизатора, реализующему протокол IP. Протокол IP извлекает из пакета адрес назначения 142.06.13.14 и просматривает записи своей таблицы маршрутизации. Пусть маршрутизатор 1 имеет в своей таблице маршрутизации запись 142.06.0.0 135.12.0.11 2, которая говорит о том, что пакеты для сети 142.06. 0.0 нужно передавать маршрутизатору 135.12.0.11, находящемуся в сети, подключенной к порту 2 маршрутизатора 1.
7. Маршрутизатор 1 просматривает параметры порта 2 и находит, что к нему подключена сеть FDDI. Так как сеть FDDI имеет значение MTU большее, чем сеть Ethernet, то фрагментация IP-пакета не требуется. Поэтому маршрутизатор 1 формирует кадр формата FDDI. На этом этапе модуль IP должен определить MAC - адрес следующего маршрутизатора по известному IP-адресу 135.12.0.11. Для этого он обращается к протоколу ARP. Допустим, что нужной записи в кэш-таблице не оказалось, тогда в сеть FDDI отправляется широковещательный ARP-запрос, содержащий наряду с прочей следующую информацию.



Порт 1 маршрутизатора 2 распознает свой IP-адрес и посылает ARP-ответ по адресу запросившего узла:



Теперь, зная MAC - адрес следующего маршрутизатора 00E0F77F51A0, маршрутизатор 1 отсылает кадр FDDI по направлению к маршрутизатору 2. Заметим, что в поле IP-адреса назначения никаких изменений не произошло.

8. Аналогично действует модуль IP на маршрутизаторе 2. Получив кадр FDDI, он отбрасывает его заголовок, а из заголовка IP извлекает IP-адрес сети назначения и просматривает свою таблицу маршрутизации. Там он может найти запись о конкретной сети назначения:

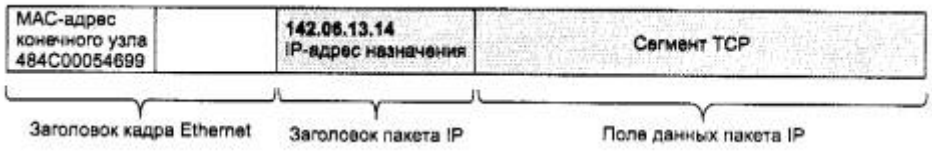
   142.06.0.0 203.21.4.12 2

   или при отсутствии такой записи будет использована запись о маршрутизаторе по умолчанию:

   default 203.21.4.12 2.

Определив IP-адрес следующего маршрутизатора 203.21.4.12, модуль IP формирует кадр Ethernet для передачи пакета маршрутизатору 3 по сети Ethernet. С помощью протокола ARP он находит MAC - адрес этого маршрутизатора и помещает его в заголовок кадра. IP-адрес узла назначения, естественно, остается неизменным.

9. Наконец, после того как пакет поступил в маршрутизатор сети назначения - маршрутизатор 3, - появляется возможность передачи этого пакета компьютеру назначения. Маршрутизатор 3 определяет, что пакет нужно передать в сеть 142.06.0,0, которая непосредственно подключена к его первому порту. Поэтому он посылает ARP-запрос по сети Ethernet с IP-адресом компьютера sl.msk.su. ARP-ответ содержит MAC - адрес конечного узла, который модуль IP передает канальному протоколу для формирования кадра Ethernet:



10. Сетевой адаптер компьютера sl.msk.su захватывает кадр Ethernet, обнаруживает совпадение MAC - адреса, содержащегося в заголовке, со своим собственным адресом и направляет его модулю IP. После анализа полей IP-заголовка из пакета извлекаются данные, которые в свою очередь содержат сообщение выше лежащего протокола. Поскольку в данном примере рассматривается обмен данными по протоколу FTP, который использует в качестве транспортного протокола TCP, то в поле данных IP-пакета находится TCP - сегмент. Определив из TCP-заголовка номер порта, модуль IP переправляет сегмент в соответствующую очередь, из которой данный сегмент попадет программному модулю FTP-сервера.

5.3.5. Маршрутизация с использованием масок

Использование масок для структуризации сети

Алгоритм маршрутизации усложняется, когда в систему адресации узлов вносятся дополнительные элементы - маски. В чем же причина отказа от хорошо себя зарекомендовавшего в течение многих лет метода адресации, основанного на классах? Таких причин несколько, и одна из них - потребность в структуризации сетей.

Часто администраторы сетей испытывают неудобства из-за того, что количество централизованно выделенных им номеров сетей недостаточно для того, чтобы структурировать сеть надлежащим образом, например разместить все слабо взаимодействующие компьютеры по разным сетям. В такой ситуации возможны два пути. Первый из них связан с получением от InterNIC или поставщика услуг Internet дополнительных номеров сетей. Второй способ, употребляющийся чаще, связан с использованием технологии масок, которая позволяет разделять одну сеть на несколько сетей.

Допустим, администратор получил в свое распоряжение адрес класса B: 129.44.0.0. Он может организовать сеть с большим числом узлов, номера которых он может брать из диапазона 0.0.0.1-0.0.255.254 (с учетом того, что адреса из одних нулей и одних единиц имеют специальное назначение и не годятся для адресации узлов). Однако ему не нужна одна большая неструктурированная сеть, производственная необходимость диктует администратору другое решение, в соответствии с которым сеть должна быть разделена на три отдельных подсети, при этом трафик в каждой подсети должен быть надежно локализован. Это позволит легче диагностировать сеть и проводить в каждой из подсетей особую политику безопасности.

Посмотрим, как решается эта проблема путем использования механизма масок.

Итак, номер сети, который администратор получил от поставщика услуг, - 129.44.0.0 (10000001 00101100 00000000 00000000). В качестве маски было выбрано значение 255.255.192.0 (11111111111111111111000000 00000000). После наложения маски на этот адрес число разрядов, интерпретируемых как номер сети, увеличилось с 16 (стандартная длина поля номера сети для класса B) до 18 (число единиц в маске), то есть администратор получил возможность использовать для нумерации подсетей два дополнительных бита. Это позволяет ему сделать из одного, централизованно заданного ему номера сети, четыре:

129.44.0.0 (10000001 00101100 00000000 00000000)

129.44.64.0 (10000001 00101100 01000000 00000000)

129.44.128.0 (10000001 00101100 10000000 00000000)

129.44.192.0 (10000001 00101100 11000000 00000000)

Два дополнительных последних бита в номере сети часто интерпретируются как номера подсетей (subnet), и тогда четыре перечисленных выше подсети имеют номера 0 (00), 1 (01), 2 (10) и 3 (11) соответственно.

ПРИМЕЧАНИЕ Некоторые программные и аппаратные маршрутизаторы не поддерживают номера подсетей, которые состоят либо только из одних нулей, либо только из одних единиц. Например, для некоторых типов оборудования номер сети 129.44.0.0 с маской 255.255.192.0, использованный в нашем примере, окажется недопустимым, поскольку в этом случае разряды в поле номера подсети имеют значение 00. По аналогичным соображениям недопустимым может оказаться и номер сети 129.44.192.0 с тем же значением маски. Здесь номер подсети состоит только из единиц. Однако более современные маршрутизаторы свободны от этих ограничений. Поэтому, принимая решение об использовании механизма масок, необходимо выяснить характеристики того оборудования, которым вы располагаете, чтобы соответствующим образом сконфигурировать маршрутизаторы и компьютеры сети.

В результате использования масок была предложена следующая схема распределения адресного пространства (рис. 5.15).

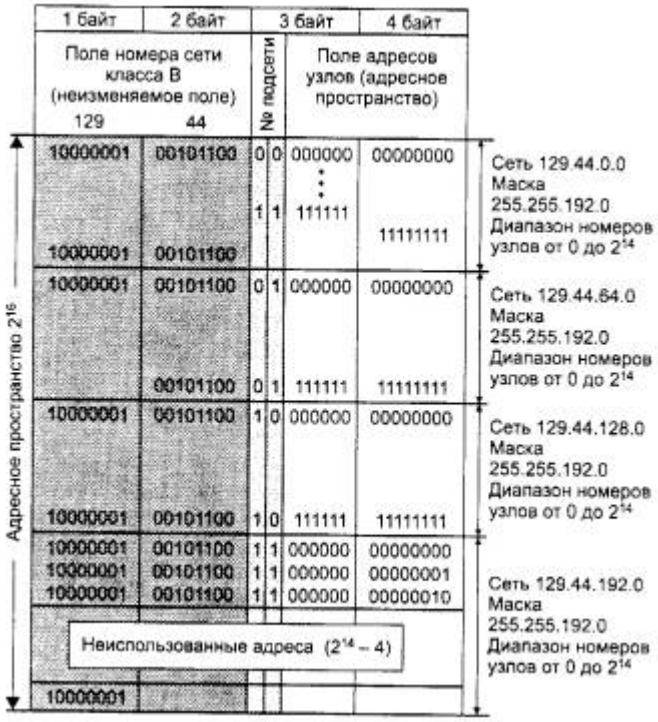


Рис. 5.15. Разделение адресного пространства сети класса B 129.44.0.0 на четыре равные части путем использования масок одинаковой длины 255.255.192.0

Сеть, получившаяся в результате проведенной структуризации, показана на рис. 5.16. Весь трафик во внутреннюю сеть 129.44.0.0, направляемый из внешней сети, поступает через маршрутизатор Ml. В целях структуризации информационных потоков во внутренней сети установлен дополнительный маршрутизатор М2.

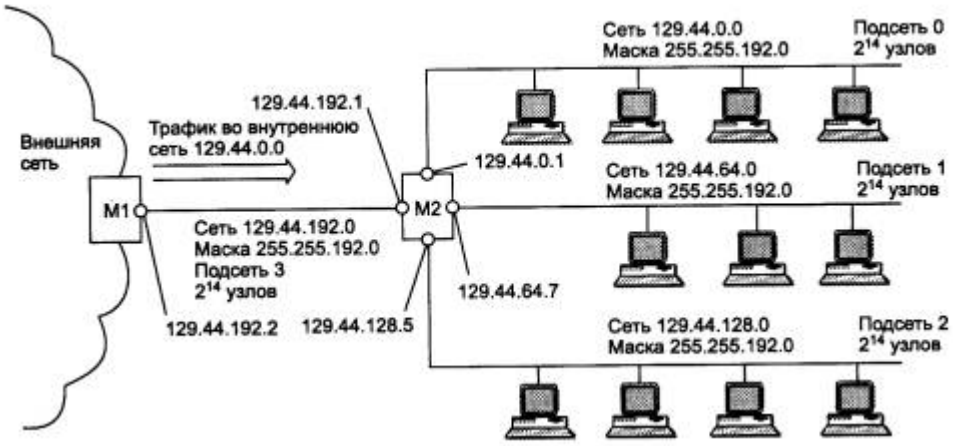


Рис. 5.16. Маршрутизация с использованием масок одинаковой длины

Все узлы были распределены по трем разным сетям, которым были присвоены номера 129.44.0.0, 129.44.64.0 и 129.44.128.0 и маски одинаковой длины - 255.255.192.0. Каждая из вновь образованных сетей была подключена к соответственно сконфигурированным портам внутреннего маршрутизатора М2. Кроме того, еще одна сеть (номер 129.44.192.0, маска 255.255.192.0) была выделена для создания соединения между внешним и внутренним маршрутизаторами. Особо отметим, что в этой сети для адресации узлов

были заняты всего два адреса 129.44.192.1 (порт маршрутизатора М2) и 129.44.192.2 (порт маршрутизатора Ml), еще два адреса 129.44.192.0 и 129.44.192.255 являются особыми адресами. Следовательно, огромное число узлов (2^{14} - 4) в этой подсети никак не используются.

Извне сеть по-прежнему выглядит, как единая сеть класса В, а на местном уровне это полноценная составная сеть, в которую входят три отдельные сети. Приходящий общий трафик разделяется местным маршрутизатором М2 между этими сетями в соответствии с таблицей маршрутизации. (Заметим, что разделение большой сети, имеющей один адрес старшего класса, например А или В, с помощью масок несет в себе еще одно преимущество по сравнению с использованием нескольких адресов стандартных классов для сетей меньшего размера, например С. Оно позволяет скрыть внутреннюю структуру сети предприятия от внешнего наблюдения и тем повысить ее безопасность.)

Рассмотрим, как изменяется работа модуля IP, когда становится необходимым учитывать наличие масок. Во-первых, в каждой записи таблицы маршрутизации появляется новое поле - поле маски.

Во-вторых, меняется алгоритм определения маршрута по таблице маршрутизации. После того как IP-адрес извлекается из очередного полученного IP-пакета, необходимо определить адрес следующего маршрутизатора, на который надо передать пакет с этим адресом. Модуль IP последовательно просматривает все записи таблицы маршрутизации. С каждой записью производятся следующие действия.

- Маска М, содержащаяся в данной записи, накладывается на IP-адрес узла назначения, извлеченный из пакета.
- Полученное в результате число является номером сети назначения обрабатываемого пакета. Оно сравнивается с номером сети, который помещен в данной записи таблицы маршрутизации.
- Если номера сетей совпадают, то пакет передается маршрутизатору, адрес которого помещен в соответствующем поле данной записи.

Теперь рассмотрим этот алгоритм на примере маршрутизации пакетов в сети, изображенной на рис. 5.16. Все маршрутизаторы внешней сети, встретив пакеты с адресами, начинающимися с 129.44, интерпретируют их как адреса класса В и направляют по маршрутам, ведущим к маршрутизатору Ml. Маршрутизатор Ml в свою очередь направляет весь входной трафик сети 129.44.0.0 на маршрутизатор М2, а именно на его порт 129.44.192.1.

Маршрутизатор М2 обрабатывает все поступившие на него пакеты в соответствии с таблицей маршрутизации (табл. 5.12).

Таблица 5.12. Таблица маршрутизатора М2 в сети с масками одинаковой длины

Номер сети	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.192.0	129.44.0.1	129.44.192.2	Подключена
129.44.64.0	255.255.192.0	129.44.64.7	129.44.64.7	Подключена
129.44.128.0	255.255.192.0	129.44.128.5	129.44.128.5	Подключена
129.44.192.0	255.255.192.0	129.44.192.1	129.44.192.1	Подключена
0.0.0.0	0.0.0.0	129.44.192.2	129.44.192.1	—

Первые четыре записи в таблице соответствуют внутренним подсетям, непосредственно подключенным к портам маршрутизатора М2.

Запись 0.0.0.0 с маской 0.0.0.0 соответствует маршруту по умолчанию. Действительно, любой адрес в пришедшем пакете после наложения на него маски 0.0.0.0 даст адрес сети 0.0.0.0, что совпадает с адресом, указанным в записи. Маршрутизатор выполняет сравнение с адресом 0.0.0.0 в последнюю очередь, в том случае когда пришедший адрес не дал совпадения ни с одной записью в таблице, отличающейся от 0.0.0.0. Записей с адресом 0.0.0.0 в таблице маршрутизации может быть несколько. В этом случае маршрутизатор передает пакет по всем таким маршрутам.

Пусть, например, с маршрутизатора Ml на порт 129.44.192.1 маршрутизатора М2 поступает пакет с адресом назначения 129.44.78.200. Модуль IP начинает последовательно просматривать все строки таблицы, до тех пор пока не найдет совпадения номера сети в адресе назначения и в строке таблицы. Маска из первой строки 255.255.192.0 накладывается на адрес 129.44,78.200, в результате чего получается номер сети 129.44.64.0.

В двоичном виде эта операция выглядит следующим образом:

10000001.00101100.01001110.11001000

11111111.11111111.11000000.00000000

- -

10000001.00101100.01000000.00000000

Полученный номер 129.44.64.0 сравнивается с номером сети в первой строке таблицы 129.44.0.0. Поскольку они не совпадают, то происходит переход к следующей строке. Теперь извлекается маска из второй строки (в данном случае она имеет такое же значение, но в общем случае это совсем не обязательно) и накладывается на адрес назначения пакета 129.44.78.200. Понятно, что из-за совпадения длины масок будет получен тот же номер сети 129.44.64.0. Этот номер совпадает с номером сети во второй строке таблицы, а значит, найден маршрут для данного пакета - он должен быть отправлен на порт маршрутизатора 129.44.64.7 в сеть, непосредственно подключенную к данному маршрутизатору.

Вот еще пример. IP-адрес 129.44.141.15(10000001 00101100 10001101 00001111), который при использовании классов делится на номер сети 129.44.0.0 и номер узла 0.0.141.15, теперь, при использовании маски 255.255.192.0, будет интерпретироваться как пара: 129.44.128.0 - номер сети, 0.0.13.15 - номер узла.

Использование масок переменной длины

В предыдущем примере использования масок (см. рис. 5.15 и 5.16) все подсети имеют одинаковую длину поля номера сети - 18 двоичных разрядов, и, следовательно, для нумерации узлов в каждой из них отводится по 14 разрядов. То есть все сети являются очень большими и имеют одинаковый размер. Однако в этом случае, как и во многих других, более эффективным явилось бы разбиение сети на подсети разного размера. В частности, большое число узлов, вполне желательное для пользовательской подсети, явно

является избыточным для подсети, которая связывает два маршрутизатора по схеме «точка-точка». В этом случае требуются всего два адреса для адресации двух портов соседних маршрутизаторов. В предыдущем же примере для этой вспомогательной сети Ml - M2 был использован номер, позволяющий адресовать 2^{14} узлов, что делает такое решение неприемлемо избыточным. Администратор может более рационально распределить имеющееся в его распоряжении адресное пространство с помощью масок переменной длины.

На рис. 5.17 приведен пример распределения адресного пространства, при котором избыточность имеющегося множества IP-адресов может быть сведена к минимуму. Половина из имеющихся адресов (215) была отведена для создания сети с адресом 129.44.0.0 и маской 255.255.128.0. Следующая порция адресов, составляющая четверть всего адресного пространства (2^{14}), была назначена для сети 129.44.128.0 с маской 255.255.192.0. Далее в пространстве адресов был «вырезан» небольшой фрагмент для создания сети, предназначенной для связывания внутреннего маршрутизатора M2 с внешним маршрутизатором Ml.

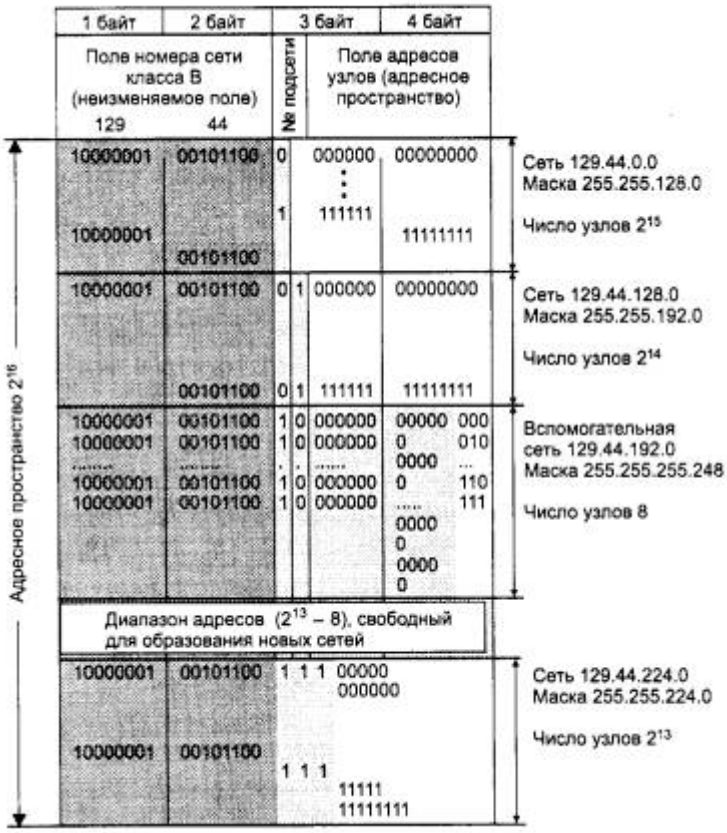


Рис. 5.17. Разделение адресного пространства сети класса B 129.44.0.0 на сети разного размера путем использования масок переменной длины

В IP-адресе такой вырожденной сети для поля номера узла как минимум должны быть отведены два двоичных разряда. Из четырех возможных комбинаций номеров узлов: 00, 01,10 и 11 два номера имеют специальное назначение и не могут быть присвоены узлам, но оставшиеся два 10 и 01 позволяет адресовать порты маршрутизаторов. В нашем примере сеть была выбрана с некоторым запасом - на 8 узлов. Поле номера узла в таком случае имеет 3 двоичных разряда, маска в десятичной нотации имеет вид 255.255.255.248, а номер сети, как видно из рис. 5.17, равен в данном конкретном случае 129.44.192.0. Если

эта сеть является локальной, то на ней могут быть расположены четыре узла помимо двух портов маршуртизаторов.

ПРИМЕЧАНИЕ Заметим, что глобальным связям между маршрутизаторами типа «точка-точка» не обязательно давать IP-адреса, так как к такой сети не могут подключаться никакие другие узлы, кроме двух портов маршрутизаторов. Однако чаще всего такой вырожденной сети все же дают IP-адрес. Это делается, например, для того, чтобы скрыть внутреннюю структуру сети и обращаться к ней по одному адресу входного порта маршрутизатора, в данном примере по адресу 129.44.192.1. Кроме того, этот адрес может понадобиться при туннелировании немаршрутизируемых протоколов в IP-пакеты, что будет рассмотрено ниже.

Оставшееся адресное пространство администратор может «нарезать» на разное количество сетей разного объема в зависимости от своих потребностей. Из оставшегося пула (2^{14} - 4) адресов администратор может образовать еще одну достаточно большую сеть с числом узлов 2^{13}. При этом свободными останутся почти столько же адресов (2^{13} - 4), которые также могут быть использованы для создания новых сетей. К примеру, из этого «остатка» можно образовать 31 сеть, каждая из которых равна размеру стандартной сети класса C, и к тому же еще несколько сетей меньшего размера. Ясно, что разбиение может быть другим, но в любом случае с помощью масок переменного размера администратор всегда имеет возможность гораздо рациональнее использовать все имеющиеся у него адреса.

На рис. 5.18 показана схема сети, структурированной с помощью масок переменной длины.

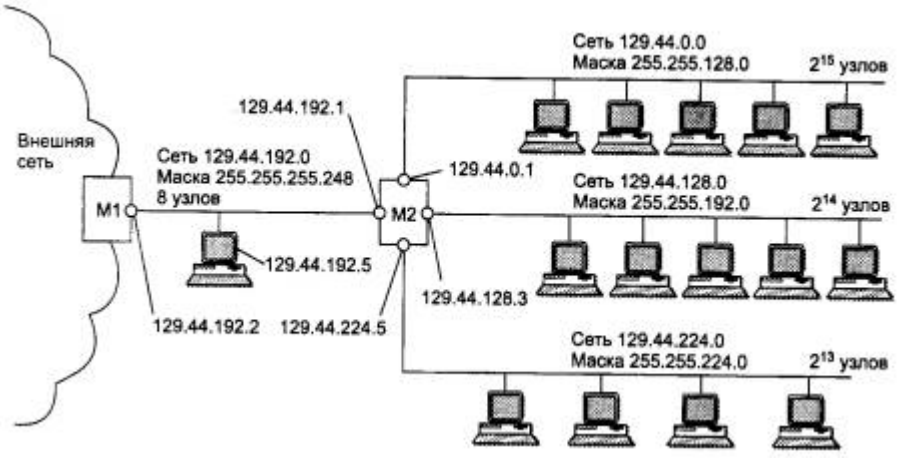


Рис. 5.18. Сеть, структурированная с использованием масок переменной длины

Таблица маршрутизации M2, соответствующая структуре сети, показанной на рис. 5.18, содержит записи о четырех непосредственно подключенных сетях и запись о маршрутизаторе по умолчанию (табл. 5.13). Процедура поиска маршрута при использовании масок переменной длины ничем не отличается от подобной процедуры, описанной ранее для масок одинаковой длины.

Таблица 5.13. Таблица маршрутизатора М2 в сети с масками переменной длины

Номер сети	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.128.0	129.44.0.1	129.44.0.1	Подключена
129.44.128.0	255.255.192.0	129.44.128.3	129.44.128.3	Подключена
129.44.192.0	255.255.255.248	129.44.192.1	129.44.191.1	Подключена
129.44.224.0	255.255.224.0	129.44.224.5	129.44.224.5	Подключена
0.0.0.0	0.0.0.0	129.44.192.2	129.44.192.1	—

Некоторые особенности масок переменной длины проявляются при наличии так называемых «перекрытий». Под перекрытием понимается наличие нескольких маршрутов к одной и той же сети или одному и тому же узлу. В этом случае адрес сети в пришедшем пакете может совпасть с адресами сетей, содержащихся сразу в нескольких записях таблицы маршрутизации.

Рассмотрим пример. Пусть пакет, поступивший из внешней сети на маршрутизатор Ml, имеет адрес назначения 129.44.192.5. Ниже приведен фрагмент таблицы маршрутизации маршрутизатора Ml. Первая из приведенных двух записей говорит о том, что все пакеты, адреса которых начинаются на 129.44, должны быть переданы на маршрутизатор М2. Эта запись выполняет *агрегирование* адресов всех подсетей, созданных на базе одной сети 129.44.0.0. Вторая строка говорит о том, что среди всех возможных подсетей сети 129.44.0.0 есть одна, 129.44.192.0, для которой пакеты можно направлять непосредственно, а не через маршрутизатор М2.

Номер сети	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
.....	.....	.....	.....	.....
129.44.0.0	255.255.0.0	129.44.192.1	129.44.191.2	2
129.44.192.0	255.255.255.248	129.44.192.2	129.44.192.2	Подключена
.....	.....	.....	.....	.....

Если следовать стандартному алгоритму поиска маршрута по таблице, то сначала на адрес назначения 129.44.192.5 накладывается маска из первой строки 255.255.0.0 и получается результат 129.44.0.0, который совпадает с номером сети в этой строке. Но и при наложении на адрес 129.44.192.5 маски из второй строки 255.255.255.248 полученный результат 129.44.192.0 также совпадает с номером сети во второй строке. В таких случаях должно быть применено следующее правило: «Если адрес принадлежит нескольким подсетям в базе данных маршрутов, то продвигающий пакет маршрутизатор использует наиболее специфический маршрут, то есть выбирается адрес подсети, дающий большее совпадение разрядов».

В данном примере будет выбран второй маршрут, то есть пакет будет передан в непосредственно подключенную сеть, а не пойдет кружным путем через маршрутизатор М2.

Механизм выбора самого специфического маршрута является обобщением понятия «маршрут по умолчанию». Поскольку в традиционной записи для маршрута по умолчанию 0.0.0.0 маска 0.0.0.0 имеет нулевую длину, то этот маршрут считается самым

неспецифическим и используется только при отсутствии совпадений со всеми остальными записями из таблицы маршрутизации.

ПРИМЕЧАНИЕ В IP-пакетах при использовании механизма масок по-прежнему передается только IP-адрес назначения, а маска сети назначения не передается. Поэтому из IP-адреса пришедшего пакета невозможно выяснить, какая часть адреса относится к номеру сети, а какая - к номеру узла. Если маски во всех подсетях имеют один размер, то это не создает проблем. Если же для образования подсетей применяют маски переменной длины, то маршрутизатор должен каким-то образом узнавать, каким адресам сетей какие маски соответствуют. Для этого используются протоколы маршрутизации, переносящие между маршрутизаторами не только служебную информацию об адресах сетей, но и о масках, соответствующих этим номерам. К таким протоколам относятся протоколы RIPv2 и OSPF, а вот, например, протокол RIP маски не распространяет и для использования масок переменной длины не подходит.

Технология бесклассовой междоменной маршрутизации CIDR

За последние несколько лет в сети Internet многое изменилось: резко возросло число узлов и сетей, повысилась интенсивность трафика, изменился характер передаваемых данных. Из-за несовершенства протоколов маршрутизации обмен сообщениями об обновлении таблиц стал иногда приводить к сбоям магистральных маршрутизаторов из-за перегрузки при обработке большого объема служебной информации. Так, в 1994 году таблицы магистральных маршрутизаторов в Internet содержали до 70 000 маршрутов.

На решение этой проблемы была направлена, в частности, и технология *бес-классовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR)*, впервые о которой было официально объявлено в 1993 году, когда были опубликованы RFC 1517, RFC 1518, RFC 1519 и RFC 1520.

Суть технологии CIDR заключается в следующем. Каждому поставщику услуг Internet должен назначаться непрерывный диапазон в пространстве IP-адресов. При таком подходе адреса всех сетей каждого поставщика услуг имеют общую старшую часть - *префикс*, поэтому маршрутизация на магистралях Internet может осуществляться на основе префиксов, а не полных адресов сетей. Агрегирование адресов позволит уменьшить объем таблиц в маршрутизаторах всех уровней, а следовательно, ускорить работу маршрутизаторов и повысить пропускную способность Internet.

Деление IP-адреса на номер сети и номер узла в технологии CIDR происходит не на основе нескольких старших бит, определяющих класс сети (A, B или C), а на основе маски переменной длины, назначаемой поставщиком услуг. На рис. 5.19 показан пример некоторого пространства IP-адресов, которое имеется в распоряжении гипотетического поставщика услуг. Все адреса имеют общую часть в k старших разрядах - префикс. Оставшиеся n разрядов используются для дополнения неизменяемого префикса переменной частью адреса. Диапазон имеющихся адресов в таком случае составляет 2^n. Когда потребитель услуг обращается к поставщику услуг с просьбой о выделении ему некоторого количества адресов, то в имеющемся пуле адресов «вырезается» непрерывная область S1, S2, S3 или S4 соответствующего размера. Причем границы этой области

выбираются такими, чтобы для нумерации требуемого числа узлов хватило некоторого числа младших разрядов, а значения всех оставшихся (старших) разрядов было одинаковым у всех адресов данного диапазона. Таким условиям могут удовлетворять только области, размер которых кратен степени двойки, А границы выделяемого участка должны быть кратны требуемому размеру.

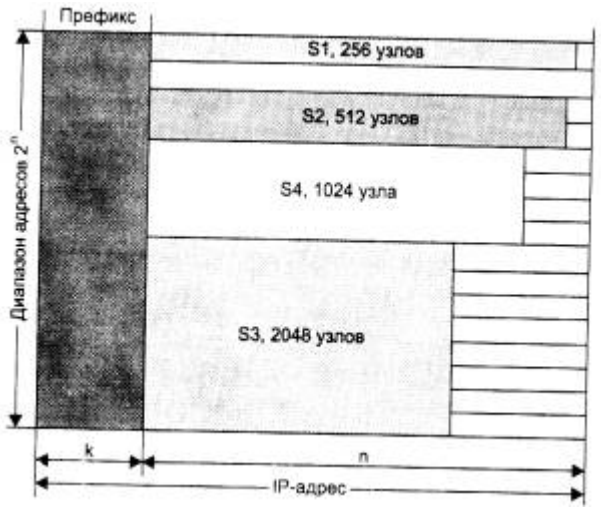


Рис. 5.19. Технологии CIDR

Рассмотрим пример. Пусть поставщик услуг Internet располагает пулом адресов в диапазоне 193.20.0.0-193.23.255.255 (1100 0001.0001 0100.0000 0000.0000 0000-11000001.0001 0111.11111111.11111111) с общим префиксом 193.20(11000001.0001 01) и маской, соответствующей этому префиксу 255.252.0.0.

Если абоненту этого поставщика услуг требуется совсем немного адресов, например 13, то поставщик мог бы предложить ему различные варианты: сеть 193.20.30.0, сеть 193.20.30.16 или сеть 193.21.204.48, все с одним и тем же значением маски 255.255.255.240. Во всех случаях в распоряжении абонента для нумерации узлов имеются 4 младших бита.

Рассмотрим другой вариант, когда к поставщику услуг обратился крупный заказчик, сам, возможно собирающийся оказывать услуги по доступу в Internet. Ему требуется блок адресов в 4000 узлов. В этом случае поставщик услуг мог бы предложить ему, например, диапазон адресов 193,22.160.0-193.22.175.255 с маской 255.255.240.0. Агрегированный номер сети (префикс) в этом случае будет равен 193.22.160.0.

Администратор маршрутизатора М2 (рис. 5.20) поместит в таблицу маршрутизации только по одной записи на каждого клиента, которому был выделен пул адресов, независимо от количества подсетей, организованных клиентом. Если клиент, получивший сеть 193.22.160.0, через некоторое время разделит ее адресное пространство в 4096 адресов на 8 подсетей, то в маршрутизаторе М2 первоначальная информация о выделенной ему сети не изменится.

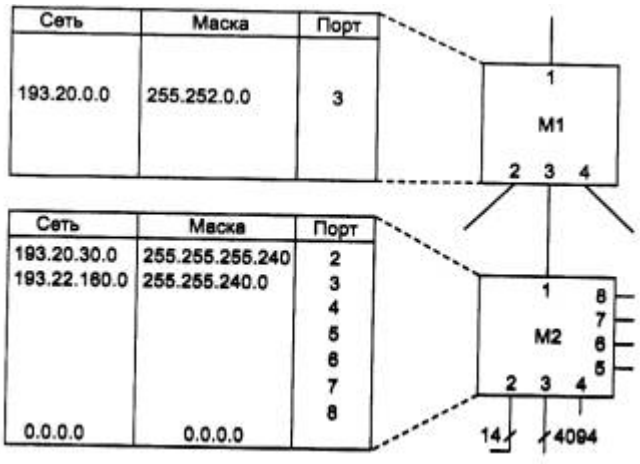


Рис. 5.20. Выигрыш в количестве записей в маршрутизаторе при использовании технологии CIDR

Для поставщика услуг верхнего уровня, поддерживающего клиентов через маршрутизатор Ml, усилия поставщика услуг нижнего уровня по разделению его адресного пространства также не будут заметны. Запись 193.20.0,0 с маской 255.252.0,0 полностью описывает сети поставщика услуг нижнего уровня в маршрутизаторе Ml.

Итак, внедрение технологии CIDR позволяет решить две основные задачи.

- Более экономное расходование адресного пространства. Действительно, получая в свое распоряжение адрес сети, например, класса С, некоторые организации не используют весь возможный диапазон адресов просто потому, что в их сети имеется гораздо меньше 255 узлов. Технология CIDR отказывается от традиционной концепции разделения адресов протокола IP на классы, что позволяет получать в пользование столько адресов, сколько реально необходимо. Благодаря технологии CIDR поставщики услуг получают возможность «нарезать» блоки из выделенного им адресного пространства в точном соответствии с требованиями каждого клиента, при этом у клиента остается пространство для маневра на случай его будущего роста.
- Уменьшение числа записей в таблицах маршрутизаторов за счет объединения маршрутов - одна запись в таблице маршрутизации может представлять большое количество сетей. Действительно, для всех сетей, номера которых начинаются с одинаковой последовательности цифр, в таблице маршрутизации может быть предусмотрена одна запись (см. рис. 5.20). Так, маршрутизатор М2 установленный в организации, которая использует технику CIDR для выделения адресов своим клиентам, должен поддерживать в своей таблице маршрутизации все 8 записей о сетях клиентов. А маршрутизатору Ml достаточно иметь одну запись о всех этих сетях, на основании которой он передает пакеты с префиксом 193.20 маршрутизатору М2, который их и распределяет по нужным портам.

Если все поставщики услуг Internet будут придерживаться стратегии CIDR, то особенно заметный выигрыш будет достигаться в магистральных маршрутизаторах.

Технология CIDR уже успешно используется в текущей версии IPv4 и поддерживается такими протоколами маршрутизации, как OSPF, RIP-2, BGP4. Предполагается, что эти же протоколы будут работать и с новой версией протокола IPv6. Следует отметить, что в

настоящее время технология CIDR поддерживается магистральными маршрутизаторами Internet, а не обычными хостами в локальных сетях.

Использование CIDR в сетях IPv4 в общем случае требует перенумерации сетей. Поскольку эта процедура сопряжена с определенными временными и материальными затратами, для ее проведения пользователей нужно каким-либо образом стимулировать. В качестве таких стимулов рассматривается, например, введение оплаты за строку в таблице маршрутизации или же за количество узлов в сети. При использовании классов сетей абонент часто не полностью занимает весь допустимый диапазон адресов узлов - 254 адреса для сети класса С или 65 534 адреса для сети класса В. Часть адресов узлов обычно пропадает. Требование оплаты каждого адреса узла поможет пользователю решиться на перенумерацию, с тем чтобы получить ровно столько адресов, сколько ему нужно.

5.3.6. Фрагментация IP-пакетов

Протокол IP позволяет выполнять фрагментацию пакетов, поступающих на входные порты маршрутизаторов.

Следует различать фрагментацию сообщений в узле-отправителе и динамическую фрагментацию сообщений в транзитных узлах сети - маршрутизаторах. Практически во всех стеках протоколов есть протоколы, которые отвечают за фрагментацию сообщений прикладного уровня на такие части, которые укладываются в кадры канального уровня. В стеке TCP/IP эту задачу решает протокол TCP, который разбивает поток байтов, передаваемый ему с прикладного уровня на сообщения нужного размера (например, на 1460 байт для протокола Ethernet). Поэтому протокол IP в узле-отправителе не использует свои возможности по фрагментации пакетов.

А вот при необходимости передать пакет в следующую сеть, для которой размер пакета является слишком большим, IP-фрагментация становится необходимой. В функции уровня IP входит разбиение слишком длинного для конкретного типа составляющей сети сообщения на более короткие пакеты с созданием соответствующих служебных полей, нужных для последующей сборки фрагментов в исходное сообщение.

В большинстве типов локальных и глобальных сетей значения MTU, то есть максимальный размер поля данных, в которое должен инкапсулировать свой пакет протокол IP, значительно отличается. Сети Ethernet имеют значение MTU, равное 1500 байт, сети FDDI - 4096 байт, а сети X.25 чаще всего работают с MTU в 128 байт.

IP-пакет может быть помечен как не фрагментируемый. Любой пакет, помеченный таким образом, не может быть фрагментирован модулем IP ни при каких условиях. Если же пакет, помеченный как не фрагментируемый, не может достигнуть получателя без фрагментации, то этот пакет просто уничтожается, а узлу-отправителю посылается соответствующее ICMP-сообщение.

Протокол IP допускает возможность использования в пределах отдельной подсети ее собственных средств фрагментирования, невидимых для протокола IP. Например, технология ATM делит поступающие IP-пакеты на ячейки с полем данных в 48 байт с помощью своего уровня сегментирования, а затем собирает ячейки в исходные пакеты на выходе из сети. Но такие технологии, как ATM, являются скорее исключением, чем правилом.

Процедуры фрагментации и сборки протокола IP рассчитаны на то, чтобы пакет мог быть разбит на практически любое количество частей, которые впоследствии могли бы быть вновь собраны. Получатель фрагмента использует поле идентификации для того, чтобы не перепутать фрагменты различных пакетов. Модуль IP, отправляющий пакет, устанавливает в поле идентификации значение, которое должно быть уникальным для данной пары отправитель - получатель, а также время, в течение которого пакет может быть активным в сети.

Поле смещения фрагмента сообщает получателю положение фрагмента в исходном пакете. Смещение фрагмента и длина определяют часть исходного пакета, принесенную этим фрагментом. Флаг «more fragments» показывает появление последнего фрагмента. Модуль протокола IP, отправляющий неразбитый на фрагменты пакет, устанавливает в нуль флаг «more fragments» и смещение во фрагменте.

Эти поля дают достаточное количество информации для сборки пакета.

Чтобы разделить на фрагменты большой пакет, модуль протокола IP, установленный, например, на маршрутизаторе, создает несколько новых пакетов и копирует содержимое полей IP-заголовка из большого пакета в IP-заголовки всех новых пакетов. Данные из старого пакета делятся на соответствующее число частей, размер каждой из которых, кроме самой последней, обязательно должен быть кратным 8 байт. Размер последней части данных равен полученному остатку.

Каждая из полученных частей данных помещается в новый пакет. Когда происходит фрагментация, то некоторые параметры IP-заголовка копируются в заголовки всех фрагментов, а другие остаются лишь в заголовке первого фрагмента. Процесс фрагментации может изменить значения данных, расположенных в поле параметров, и значение контрольной суммы заголовка, изменить значение флага «more fragments» и смещение фрагмента, изменить длину IP-заголовка и общую длину пакета, В заголовок каждого пакета заносятся соответствующие значения в поле смещения «fragment offset», а в поле общей длины пакета помещается длина каждого пакета. Первый фрагмент будет иметь в поле «fragment offset» нулевое значение. Во всех пакетах, кроме последнего, флаг «more fragments» устанавливается в единицу, а в последнем фрагменте - в нуль.

Чтобы собрать фрагменты пакета, модуль протокола IP (например, модуль на хост - компьютере) объединяет IP-пакеты, имеющие одинаковые значения в полях идентификатора, отправителя, получателя и протокола. Таким образом, отправитель должен выбрать идентификатор таким образом, чтобы он был уникален для данной пары отправитель-получатель, для данного протокола и в течение того времени, пока данный пакет (или любой его фрагмент) может существовать в составной IP-сети.

Очевидно, что модуль протокола IP, отправляющий пакеты, должен иметь таблицу идентификаторов, где каждая запись соотносится с каждым отдельным получателем, с которым осуществлялась связь, и указывает последнее значение максимального времени жизни пакета в IP-сети. Однако, поскольку поле идентификатора допускает 65 536 различных значений, некоторые хосты могут использовать просто уникальные идентификаторы, не зависящие от адреса получателя.

В некоторых случаях целесообразно, чтобы идентификаторы IP-пакетов выбирались протоколами более высокого, чем IP, уровня. Например, в протоколе TCP предусмотрена повторная передача TCP - сегментов, по каким-либо причинам не дошедшим до адресата.

Вероятность правильного приема увеличивалась бы, если бы при повторной передаче идентификатор для IP-пакета был бы тем же, что и в исходном IP-пакете, поскольку его фрагменты могли бы использоваться для сборки правильного TCP - сегмента.

Процедура объединения заключается в помещении данных из каждого фрагмента в позицию, указанную в заголовке пакета в поле «fragment offset».

Каждый модуль IP должен быть способен передать пакет из 68 байт без дальнейшей фрагментации. Это связано с тем, что IP-заголовок может включать до 60 байт, а минимальный фрагмент данных - 8 байт. Каждый получатель должен быть в состоянии принять пакет из 576 байт в качестве единого куска либо в виде фрагментов, подлежащих сборке.

Если бит флага запрета фрагментации (Don't Fragment, DF) установлен, то фрагментация данного пакета запрещена, даже если в этом случае он будет потерян. Данное средство может использоваться для предотвращения фрагментации в тех случаях, когда хост - получатель не имеет достаточных ресурсов для сборки фрагментов.

Работа протокола IP по фрагментации пакетов в хостах и маршрутизаторах иллюстрируется на рис. 5.21.

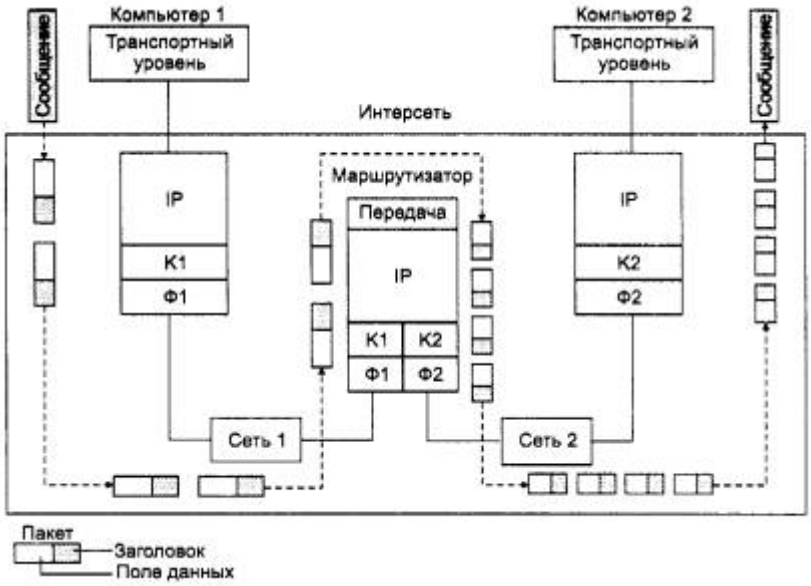


Рис.5.21. Фрагментация IP-пакетов при передаче между сетями с разным максимальным размером пакетов: К1 и 01 - канальный и физический уровень сети 1; К2 и Ф2 - канальный и физический уровень сети 2

Пусть компьютер 1 связан с сетью, имеющей значение MTU в 4096 байт, например с сетью FDDI, При поступлении на IP-уровень компьютера 1 сообщения от транспортного уровня размером в 5600 байт протокол IP делит его на два IP-пакета, устанавливая в первом пакете признак фрагментации и присваивая пакету уникальный идентификатор, например 486, В первом пакете величина поля смещения равна 0, а во втором - 2800. Признак фрагментации во втором пакете равен нулю, что показывает, что это последний фрагмент пакета. Общая величина IP-пакета составляет 2800 плюс 20 (размер IP-заголовка), то есть 2820 байт, что умещается в поле данных кадра FDDI. Далее модуль IP компьютера 1 передает эти пакеты своему сетевому интерфейсу (образуемому

протоколами канального уровня К 1 и физического уровня Ф1), Сетевой интерфейс отправляет кадры следующему маршрутизатору.

После того, как кадры пройдут уровень сетевого интерфейса маршрутизатора (К1 и Ф1) и освободятся от заголовков FDDI, модуль IP по сетевому адресу определяет, что прибывшие два пакета нужно передать в сеть 2, которая является сетью Ethernet и имеет значение MTU, равное 1500. Следовательно, прибывшие IP-пакеты необходимо фрагментировать. Маршрутизатор извлекает поле данных из каждого пакета и делит его еще пополам, чтобы каждая часть уместилась в поле данных кадра Ethernet. Затем он формирует новые IP-пакеты, каждый из которых имеет длину 1400 + 20 - 1420 байт, что меньше 1500 байт, поэтому они нормально помещаются в поле данных кадров Ethernet.

В результате в компьютер 2 по сети Ethernet приходят четыре IP-пакета с общим идентификатором 486, что позволяет протоколу IP, работающему в компьютере 2, правильно собрать исходное сообщение. Если пакеты пришли не в том порядке, в котором были посланы, то смещение укажет правильный порядок их объединения.

Отметим, что IP-маршрутизаторы не собирают фрагменты пакетов в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться по интерсети по различным маршрутам, поэтому нет гарантии, что все фрагменты проходят через какой-либо промежуточный маршрутизатор на их пути.

При приходе первого фрагмента пакета узел назначения запускает таймер, который определяет максимально допустимое время ожидания прихода остальных фрагментов этого пакета. Таймер устанавливается на максимальное из двух значений: первоначальное установочное время ожидания и время жизни, указанное в принятом фрагменте. Таким образом, первоначальная установка таймера является нижней границей для времени ожидания при сборi. Если таймер истекает раньше прибытия последнего фрагмента, то все ресурсы сборки, связанные с данным пакетом, освобождаются, все полученные к этому моменту фрагменты пакета отбрасываются, а в узел, пославший исходный пакет, направляется сообщение об ошибке с помощью протокола ICMP.

5.3.7. Протокол надежной доставки TCP-сообщений

Протокол IP является дейтаграммным протоколом и поэтому по своей природе не может гарантировать надежность передачи данных. Эту задачу - обеспечение надежного канала обмена данными между прикладными процессами в составной сети -решает протокол TCP (Transmission Control Protocol), относящийся к транспортному уровню.

Протокол TCP работает непосредственно над протоколом IP и использует для транспортировки своих блоков данных потенциально ненадежный протокол IP. Надежность передачи данных протоколом TCP достигается за счет того, что он основан на установлении логических соединений между взаимодействующими процессами. До тех пор пока программы протокола TCP продолжают функционировать корректно, а составная сеть не распалась на несвязные части, ошибки в передаче данных на уровне протокола IP не будут влиять на правильное получение данных.

Протокол IP используется протоколом TCP в качестве транспортного средства. Перед отправкой своих блоков данных протокол TCP помещает их в оболочку IP-пакета. При необходимости протокол IP осуществляет любую фрагментацию и сборку блоков данных

TCP, требующуюся для осуществления передачи и доставки через множество сетей и промежуточных шлюзов.

На рис. 5.22 показано, как процессы, выполняющиеся на двух конечных узлах, устанавливают с помощью протокола TCP надежную связь через составную сеть, все узлы которой используют для передачи сообщений дейтаграммный протокол IP.

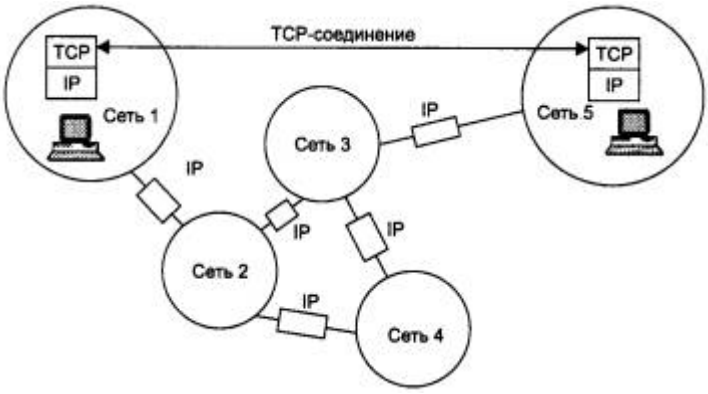


Рис. 5.22. TCP-соединение создает надежный канал связи между конечными узлами

Порты

Протокол TCP взаимодействует через межуровневые интерфейсы с ниже лежащим протоколом IP и с выше лежащими протоколами прикладного уровня или приложениями.

В то время как задачей сетевого уровня, к которому относится протокол IP, является передача данных между произвольными узлами сети, задача транспортного уровня, которую решает протокол TCP, заключается в передаче данных между любыми *прикладными процессами*, выполняющимися на любых узлах сети. Действительно, после того как пакет средствами протокола IP доставлен в компьютер-получатель, данные необходимо направить конкретному процессу-получателю. Каждый компьютер может выполнять несколько процессов, более того, прикладной процесс тоже может иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных.

Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии TCP/IP такие системные очереди называются *портами*. Таким образом, адресом назначения, который используется протоколом TCP, является идентификатор (номер) порта прикладной службы. Номер порта в совокупности с номером сети и номером конечного узла однозначно определяют прикладной процесс в сети. Этот набор идентифицирующих параметров имеет название *сокет (socket)*.

Назначение номеров портов прикладным процессам осуществляется либо *централизованно*, если эти процессы представляют собой популярные общедоступные службы (например, номер 21 закреплен за службой удаленного доступа к файлам FTP, а 23 - за службой удаленного управления telnet), либо локально для тех служб, которые еще не стали столь распространенными, чтобы закреплять за ними стандартные (зарезервированные) номера. Централизованное присвоение службам номеров портов выполняется организацией *Internet Assigned Numbers Authority (IANA)*. Эти номера затем закрепляются и опубликовываются в стандартах Internet (RFC 1700).

Локальное присвоение номера порта заключается в том, что разработчик некоторого приложения просто связывает с ним любой доступный, произвольно выбранный числовой идентификатор, обращая внимание на то, чтобы он не входил в число зарезервированных номеров портов. В дальнейшем все удаленные запросы к данному приложению от других приложений должны адресоваться с указанием назначенного ему номера порта.

Протокол TCP ведет для каждого порта две очереди: очередь пакетов, поступающих в данный порт из сети, и очередь пакетов, отправляемых данным портом в сеть. Процедура обслуживания протоколом TCP запросов, поступающих от нескольких различных прикладных служб, называется *мультиплексированием*. Обратная процедура распределения протоколом TCP поступающих от сетевого уровня пакетов между набором высокоуровневых служб, идентифицированных номерами портов, называется *демультиплексированием* (рис. 5.23).

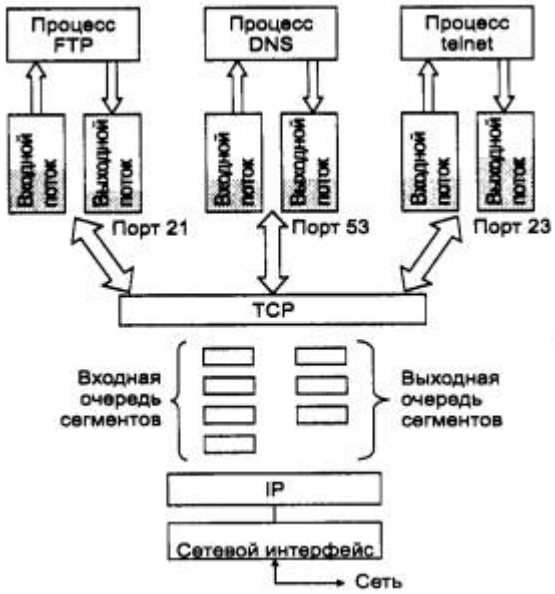


Рис. 5.23. Функции протокола TCP по мультиплексированию и демультиплексированию потоков

Сегменты и потоки

Единицей данных протокола TCP является *сегмент*. Информация, поступающая к протоколу TCP в рамках логического соединения от протоколов более высокого уровня, рассматривается протоколом TCP как неструктурированный *поток* байтов. Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая и называется сегментом (см. рис. 5.23). В отличие от многих других протоколов, протокол TCP подтверждает получение не пакетов, а байтов потока.

Не все сегменты, посланные через соединение, будут одного и того же размера, однако оба участника соединения должны договориться о максимальном размере сегмента, который они будут использовать. Этот размер выбирается таким образом, чтобы при упаковке сегмента в IP-пакет он помещался туда целиком, то есть максимальный размер сегмента не должен превосходить максимального размера поля данных IP-пакета, В

противном случае пришлось бы выполнять фрагментацию, то есть делить сегмент на несколько частей, чтобы разместить его в IP-пакете,

Соединения

Для организации надежной передачи данных предусматривается установление *логического соединения* между двумя прикладными процессами. Поскольку соединения устанавливаются через ненадежную коммуникационную систему, основанную на протоколе IP, то во избежание ошибочной инициализации соединений используется специальная многошаговая процедура подтверждения связи.

Соединение в протоколе TCP идентифицируется парой полных адресов обоих взаимодействующих процессов - сокетов. Каждый из взаимодействующих процессов может участвовать в нескольких соединениях.

Формально соединение можно определить как набор параметров, характеризующий процедуру обмена данными между двумя процессами. Помимо полных адресов процессов этот набор включает и параметры, значения которых определяются в результате переговорного процесса модулей TCP двух сторон соединения. К таким параметрам относятся, в частности, согласованные размеры сегментов, которые может посылать каждая из сторон, объемы данных, которые разрешено передавать без получения на них подтверждения, начальные и текущие номера передаваемых байтов. Некоторые из этих параметров остаются постоянными в течение всего сеанса связи, а некоторые адаптивно изменяются.

В рамках соединения осуществляется обязательное подтверждение правильности приема для всех переданных сообщений и при необходимости выполняется повторная передача. Соединение в TCP позволяет вести передачу данных одновременно в обе Стороны, то есть полнодуплексную передачу.

Реализация скользящего окна в протоколе TCP

В рамках установленного соединения правильность передачи каждого сегмента должна подтверждаться квитанцией получателя. *Квитирование* - это один из традиционных методов обеспечения надежной связи. В протоколе TCP используется частный случай квитирования - алгоритм скользящего окна. Идея этого алгоритма была изложена в главе 2, «Основы передачи дискретных данных».

Особенность использования алгоритма скользящего окна в протоколе TCP состоит в том, что, хотя единицей передаваемых данных является сегмент, окно определено на множестве нумерованных байтов неструктурированного потока данных, поступающих с верхнего уровня и буферизуемых протоколом TCP. Получающий модуль TCP отправляет «окно» посылающему модулю TCP. Данное окно задает количество байтов (начиная с номера байта, о котором уже была выслана квитанция), которое принимающий модуль TCP готов в настоящий момент принять.

Квитанция (подтверждение) посылается только в случае правильного приема данных, отрицательные квитанции не посылаются. Таким образом, отсутствие квитанции означает либо прием искаженного сегмента, либо потерю сегмента, либо потерю квитанции. В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в

которое помещает число, на единицу превышающее максимальный номер байта в полученном сегменте. Это число часто называют *номером очереди*.

На рис. 5.24 показан поток байтов, поступающий на вход протокола TCP. Из потока байтов модуль TCP нарезает последовательность сегментов. Для определенности на рисунке принято направление перемещения данных справа налево. В этом потоке можно указать несколько логических границ. Первая граница отделяет сегменты, которые уже были отправлены и на которые уже пришли квитанции. Следующую часть потока составляют сегменты, которые также уже отправлены, так как входят в границы, определенные окном, но квитанции на них пока не получены. Третья часть потока - это сегменты, которые пока не отправлены, но могут быть отправлены, так как входят в пределы окна. И наконец, последняя граница указывает на начало последовательности сегментов, ни один из которых не может быть отправлен до тех пор, пока не придет очередная квитанция и окно не будет сдвинуто вправо.

Рис. 5.24. Особенности реализации алгоритма скользящего окна в протоколе TCP

Если размер окна равен W, а последняя по времени квитанция содержала значение N, то отправитель может посылать новые сегменты до тех пор, пока в очередной сегмент не попадет байт с номером N+W. Этот сегмент выходит за рамки окна, и передачу в таком случае необходимо приостановить до прихода следующей квитанции.

Надежность передачи достигается благодаря подтверждениям и номерам очереди. Концептуально каждому байту данных присваивается номер очереди. Номер очереди для первого байта данных в сегменте передается вместе с этим сегментом и называется номером очереди для сегмента. Сегменты также несут номер подтверждения, который является номером для следующего ожидаемого байта данных, передаваемого в обратном направлении. Когда протокол TCP передает сегмент с данными, он помещает его копию в очередь повторной передачи и запускает таймер. Когда приходит подтверждение для этих данных, соответствующий сегмент удаляется из очереди. Если подтверждение не приходит до истечения срока, то сегмент посылается повторно.

Выбор времени ожидания (тайм-аута) очередной квитанции является важной задачей, результат решения которой влияет на производительность протокола TCP. Тайм-аут не должен быть слишком коротким, чтобы по возможности исключить избыточные повторные передачи, которые снижают полезную пропускную способность системы. Но он не должен быть и слишком большим, чтобы избежать длительных простоев, связанных с ожиданием несуществующей или «заблудившейся» квитанции.

При выборе величины тайм-аута должны учитываться скорость и надежность физических линий связи, их протяженность и многие другие подобные факторы. В протоколе TCP тайм-аут определяется с помощью достаточно сложного адаптивного алгоритма, идея

которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времени оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент. Практика показывает, что значение этого коэффициента должно превышать 2. В сетях с большим разбросом времени оборота при выборе тайм-аута учитывается и дисперсия этой величины.

Поскольку каждый байт пронумерован, то каждый из них может быть опознан. Приемлемый механизм опознавания является накопительным, поэтому опознавание номера X означает, что все байты с предыдущими номерами уже получены. Этот механизм позволяет регистрировать появление дубликатов в условиях повторной передачи. Нумерация байтов в пределах сегмента осуществляется так, чтобы первый байт данных сразу вслед за заголовком имел наименьший номер, а следующие за ним байты имели номера по возрастающей.

Окно, посылаемое с каждым сегментом, определяет диапазон номеров очереди, которые отправитель окна (он же получатель данных) готов принять в настоящее время. Предполагается, что такой механизм связан с наличием в данный момент места в буфере данных.

Варьируя величину окна, можно влиять на загрузку сети. Чем больше окно, тем большую порцию неподтвержденных данных можно послать в сеть. Но если пришло большее количество данных, чем может быть принято программой TCP, данные будут отброшены. Это приведет к излишним пересылкам информации и ненужному увеличению нагрузки на сеть и программу TCP.

С другой стороны, указание окна малого размера может ограничить передачу данных скоростью, которая определяется временем путешествия по сети каждого посылаемого сегмента. Чтобы избежать применения малых окон, получателю данных предлагается откладывать изменение окна до тех пор, пока свободное место не составит 20-40 % от максимально возможного объема памяти для этого соединения. Но и отправителю не стоит спешить с посылкой данных, пока окно не станет достаточно большим. Учитывая эти соображения, разработчики протокола TCP предложили схему, согласно которой при установлении соединения заявляется большое окно, но впоследствии его размер существенно уменьшается.

Если сеть не справляется с нагрузкой, то возникают очереди в промежуточных узлах - маршрутизаторах и в конечных узлах-компьютерах.

При переполнении приемного буфера конечного узла «перегруженный» протокол TCP, отправляя квитанцию, помещает в нее новый, уменьшенный размер окна. Если он совсем отказывается от приема, то в квитанции указывается окно нулевого размера. Однако даже после этого приложение может послать сообщение на отказавшийся от приема порт. Для этого сообщение должно сопровождаться пометкой «срочно». В такой ситуации порт обязан принять сегмент, даже если для этого придется вытеснить из буфера уже находящиеся там данные. После приема квитанции с нулевым значением окна протокол-отправитель время от времени делает контрольные попытки продолжить обмен данными. Если протокол-приемник уже готов принимать информацию, то в ответ на контрольный 'запрос он посылает квитанцию с указанием ненулевого размера окна.

Другим проявлением перегрузки сети является переполнение буферов в маршрутизаторах. В таких случаях они могут централизованно изменить размер окна, посылая управляющие сообщения некоторым конечным узлам, что позволяет им дифференцированно управлять интенсивностью потока данных в разных частях сети.

Выводы

- Протокол IP решает задачу доставки сообщений между узлами составной сети. Протокол IP относится к протоколам без установления соединений, поэтому он не дает никаких гарантий надежной доставки сообщений. Все вопросы обеспечения надежности доставки данных в составной сети в стеке TCP/IP решает протокол TCP, основанный на установлении логических соединений между взаимодействующими процессами.
- IP-пакет состоит из заголовка и поля данных. Максимальная длина пакета 65 535 байт, Заголовок обычно имеет длину 20 байт и содержит информацию о сетевых адресах отправителя и получателя, о параметрах фрагментации, о времени жизни пакета, о контрольной сумме и некоторых других. В поле данных IP-пакета находятся сообщения более высокого уровня, например TCP или UDP.
- Вид таблицы IP-маршрутизации зависит от конкретной реализации маршрутизатора, но, несмотря на достаточно сильные внешние различия, в таблицах всех типов маршрутизаторов есть все ключевые поля, необходимые для выполнения маршрутизации.
- Существует несколько источников, поставляющих записи в таблицу маршрутизации. Во-первых, при инициализации программное обеспечение стека TCP/ IP заносит в таблицу записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию, а также записи об особых адресах типа 127.0.0.0. Во-вторых, администратор вручную заносит статические записи о специфичных маршрутах или о маршрутизаторе по умолчанию. В-третьих, протоколы маршрутизации автоматически заносят в таблицу динамические записи о имеющихся маршрутах.
- Эффективным средством структуризации IP-сетей являются маски. Маски позволяют разделить одну сеть на несколько подсетей. Маски одинаковой длины используются для деления сети на подсети равного размера, а маски переменной длины - для деления сети на подсети разного размера. Использование масок модифицирует алгоритм маршрутизации, поэтому в этом случае предъявляются особые требования к протоколам маршрутизации в сети, к техническим характеристикам маршрутизаторов и процедурам их конфигурирования.
- Значительная роль в будущем IP-сетей отводится технологии бесклассовой междоменной маршрутизации (CIDR), которая решает две основные задачи. Первая состоит в более экономном расходование адресного пространства - благодаря CIDR поставщики услуг получают возможность «нарезать» блоки разных размеров из выделенного им адресного пространства в точном соответствии с требованиями каждого клиента. Вторая задача заключается в уменьшении числа записей в таблицах маршрутизации за счет объединения маршрутов - одна запись в таблице маршрутизации может представлять большое количество сетей с общим префиксом.
- Важной особенностью протокола IP, отличающей его от других сетевых протоколов, является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными MTU. Это свойство во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

5.4. Протоколы маршрутизации в IP-сетях

5.4.1. Внутренние и внешние протоколы маршрутизации Internet

Большинство протоколов маршрутизации, применяемых в современных сетях с коммутацией пакетов, ведут свое происхождение от сети Internet и ее предшественницы - сети ARPANET. Для того чтобы понять их назначение и особенности, полезно сначала познакомиться со структурой сети Internet, которая наложила отпечаток на терминологию и типы протоколов.

Internet изначально строилась как сеть, объединяющая большое количество существующих систем. С самого начала в ее структуре выделяли *магистральную сеть (core backbone network)*, а сети, присоединенные к магистрали, рассматривались как *автономные системы (autonomous systems, AS)*. Магистральная сеть и каждая из автономных систем имели свое собственное административное управление и собственные протоколы маршрутизации. Необходимо подчеркнуть, что автономная система и домен имен Internet - это разные понятия, которые служат разным целям. Автономная система объединяет сети, в которых под общим административным руководством одной организации осуществляется маршрутизация, а домен объединяет компьютеры (возможно, принадлежащие разным сетям), в которых под общим административным руководством одной организации осуществляется назначение уникальных символьных имен. Естественно, области действия автономной системы и домена имен могут в частном случае совпадать, если одна организация выполняет обе указанные функции.

Общая схема архитектуры сети Internet показана на рис. 5.25. Далее маршрутизаторы мы будем называть шлюзами, чтобы оставаться в русле традиционной терминологии Internet.

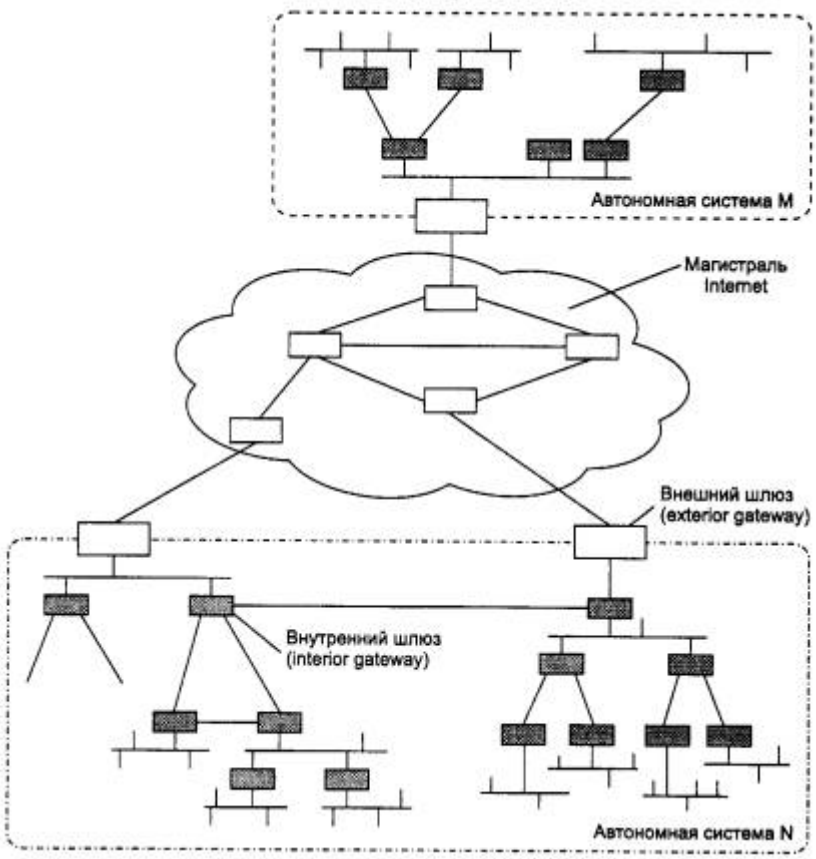


Рис. 5.25. Магистраль и автономные системы Internet

Шлюзы, которые используются для образования сетей и подсетей внутри автономной системы, называются *внутренними шлюзами (interior gateways),* а шлюзы, с помощью которых автономные системы присоединяются к магистрали сети, называются *внешними шлюзами (exterior gateways).* Магистраль сети также является автономной системой. Все автономные системы имеют уникальный 16-разрядный номер, который выделяется организацией, учредившей новую автономную систему, InterNIC.

Соответственно протоколы маршрутизации внутри автономных систем называются *протоколами внутренних шлюзов (interior gateway protocol, IGP),* а протоколы, определяющие обмен маршрутной информацией между внешними шлюзами и шлюзами магистральной сети - *протоколами внешних шлюзов (exterior gateway protocol, EGP).* Внутри магистральной сети также допустим любой собственный внутренний протокол IGP.

Смысл разделения всей сети Internet на автономные системы - в ее многоуровневом модульном представлении, что необходимо для любой крупной системы, способной к расширению в больших масштабах. Изменение протоколов маршрутизации внутри какой-либо автономной системы никак не должно влиять на работу остальных автономных систем. Кроме того, деление Internet на автономные системы должно способствовать агрегированию информации в магистральных и внешних шлюзах. Внутренние шлюзы могут использовать для внутренней маршрутизации достаточно подробные графы связей между собой, чтобы выбрать наиболее рациональный маршрут. Однако если информация такой степени детализации будет храниться во всех маршрутизаторах сети, то топологические базы данных так разрастутся, что потребуют наличия памяти гигантских размеров, а время принятия решений о маршрутизации станет неприемлемо большим.

Поэтому детальная топологическая информация остается внутри автономной системы, а автономную систему как единое целое для остальной части Internet представляют внешние шлюзы, которые сообщают о внутреннем составе автономной системы минимально необходимые сведения - количество IP-сетей, их адреса и внутреннее расстояние до этих сетей от данного внешнего шлюза.

Техника бесклассовой маршрутизации CIDR может значительно сократить объемы маршрутной информации, передаваемой между автономными системами. Так, если все сети внутри некоторой автономной системы начинаются с общего префикса, например 194.27.0.0/16, то внешний шлюз этой автономной системы должен делать объявления только об этом адресе, не сообщая отдельно о существовании внутри данной автономной системы, например, сети 194.27.32.0/19 или 194.27.40.0/21, так как эти адреса агрегируются в адрес 194.27.0.0/16.

Приведенная на рис. 5.25 структура Internet с единственной магистралью достаточно долго соответствовала действительности, поэтому специально для нее был разработан протокол обмена маршрутной информации между автономными системами, названный EGP. Однако по мере развития сетей поставщиков услуг структура Internet стала гораздо более сложной, с произвольным характером связей между автономными системами. Поэтому протокол EGP уступил место протоколу BGP, который позволяет распознать наличие петель между автономными системами и исключить их из межсистемных маршрутов. Протоколы EGP и BGP используются только во внешних шлюзах автономных систем, которые чаще всего организуются поставщиками услуг Internet. В

маршрутизаторах корпоративных сетей работают внутренние протоколы маршрутизации, такие как RIP и OSPF.

5.4.2. Дистанционно-векторный протокол RIP

Построение таблицы маршрутизации

Протокол RIP (Routing Information Protocol) является внутренним протоколом маршрутизации дистанционно-векторного типа, он представляет собой один из наиболее ранних протоколов обмена маршрутной информацией и до сих пор чрезвычайно распространен в вычислительных сетях ввиду простоты реализации. Кроме версии RIP для сетей TCP/IP существует также версия RIP для сетей IPX/SPX компании Novell.

Для IP имеются две версии протокола RIP: первая и вторая. Протокол RIPvl не поддерживает масок, то есть он распространяет между маршрутизаторами только информацию о номерах сетей и расстояниях до них, а информацию о масках этих сетей не распространяет, считая, что все адреса принадлежат к стандартными классам A, B или C. Протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня. Так как при построении таблиц маршрутизации работа версии 2 принципиально не отличается от версии 1, то в дальнейшем для упрощения записей будет описываться работа первой версии.

В качестве расстояния до сети стандарты протокола RIP допускают различные виды метрик: хопы, метрики, учитывающие пропускную способность, вносимые задержки и надежность сетей (то есть соответствующие признакам D, T и R в поле «Качество сервиса» IP-пакета), а также любые комбинации этих метрик. Метрика должна обладать свойством аддитивности - метрика составного пути должна быть равна сумме метрик составляющих этого пути. В большинстве реализации RIP используется простейшая метрика - количество хопов, то есть количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP на примере составной сети, изображенной на рис. 5.26.

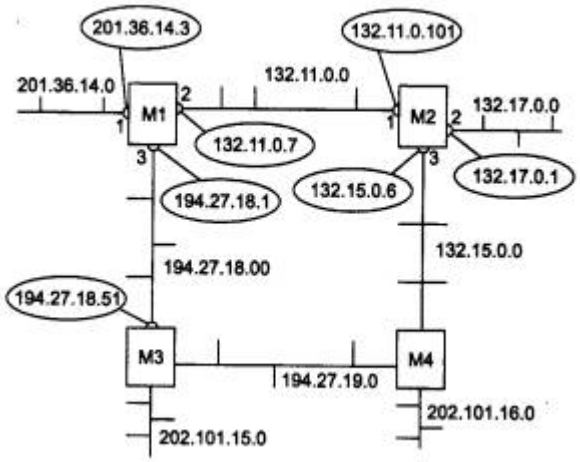


Рис. 5.26. Сеть, объединенная RIP-маршрутизаторами

Этап 1 - создание минимальных таблиц

В этой сети имеется восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами: Ml, М2, М3 и М4. Маршрутизаторы, работающие по протоколу RIP, могут иметь идентификаторы, однако для работы протокола они не являются необходимыми. В RIP-сообщениях эти идентификаторы не передаются.

В исходном состоянии в каждом маршрутизаторе программным обеспечением стека TCP/IP автоматически создается минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединенные сети. На рисунке адреса портов маршрутизаторов в отличие от адресов сетей помещены в овалы.

Таблица 5.14 позволяет оценить примерный вид минимальной таблицы маршрутизации маршрутизатора Ml.

Таблица 5.14. Минимальная таблица маршрутизации маршрутизатора Ml

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1

Минимальные таблицы маршрутизации в других маршрутизаторах будут выглядеть соответственно, например, таблица маршрутизатора М2 будет состоять из трех записей (табл. 5.15).

Таблица 5.15. Минимальная таблица маршрутизации маршрутизатора М2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
132.11.0.0	132.11.0.101	1	1
132.17.0.0	132.17.0.1	2	1
132.15.0.0	132.15.0.6	3	1

Этап 2 - рассылка минимальных таблиц соседям

После инициализации каждого маршрутизатора он начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица.

RIP-сообщения передаются в пакетах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние до нее от передающего сообщение маршрутизатора.

Соседями являются те маршрутизаторы, которым данный маршрутизатор непосредственно может передать IP-пакет по какой-либо своей сети, не пользуясь услугами промежуточных маршрутизаторов. Например, для маршрутизатора Ml соседями являются маршрутизаторы М2 и М3, а для маршрутизатора М4 - маршрутизаторы М2 и М3.

Таким образом, маршрутизатор Ml передает маршрутизатору М2 и М3 следующее сообщение:

сеть 201.36.14.0, расстояние 1;

сеть 132.11.0.0, расстояние 1;

сеть 194.27.18.0, расстояние 1.

Этап 3 - получение RIP-сообщений от соседей и обработка полученной информации

После получения аналогичных сообщений от маршрутизаторов М2 и М3 маршрутизатор Ml наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора будет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации (табл. 5.16).

Таблица 5.16. Таблица маршрутизации маршрутизатора М1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
194.27.19.0	194.27.18.51	3	2
202.101.15.0	194.27.18.51	3	2
~~132.11.0.0~~	~~132.11.0.101~~	~~2~~	~~2~~
~~194.27.18.0~~	~~194.27.18.51~~	~~3~~	~~2~~

Записи с четвертой по девятую получены от соседних маршрутизаторов, и они претендуют на помещение в таблицу. Однако только записи с четвертой по седьмую попадают в таблицу, а записи восьмая и девятая - нет. Это происходит потому, что они содержат данные об уже имеющихся в таблице Ml сетях, а расстояние до них хуже, чем в существующих записях.

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (расстояние в хопах меньше), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остаётся только одна запись; если же имеется несколько равнозначных в отношении расстояния путей к одной и той же сети, то все равно в таблице остается одна запись, которая пришла в маршрутизатор первая по времени. Для этого правила существует исключение - если худшая информация о какой-либо сети пришла от того же маршрутизатора, на основании сообщения которого была создана данная запись, то худшая информация замещает лучшую.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4 - рассылка новой, уже не минимальной, таблицы соседям

Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные о всех известных ему сетях - как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5 - получение RIP-сообщений от соседей и обработка полученной информации

Этап 5 повторяет этап 3 - маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации.

Посмотрим, как это делает маршрутизатор Ml (табл. 5.17).

Таблица 5.17. Таблица маршрутизации маршрутизатора М1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
~~132.15.0.0~~	~~194.27.18.51~~	~~3~~	~~3~~
194.27.19.0	194.27.18.51	3	2
~~194.27.19.0~~	~~132.11.0.101~~	~~2~~	~~3~~
~~202.101.15.0~~	~~194.27.18.51~~	~~3~~	~~2~~
202.101.16.0	132.11.0.101	2	3
~~202.101.16.0~~	~~194.27.18.51~~	~~3~~	~~3~~

На этом этапе маршрутизатор Ml получил от маршрутизатора М3 информацию о сети 132.15.0.0, которую тот в свою очередь на предыдущем цикле работы получил от маршрутизатора М4. Маршрутизатор уже знает о сети 132.15.0.0, причем старая информация имеет лучшую метрику, чем новая, поэтому новая информация об этой сети отбрасывается.

О сети 202.101.16.0 маршрутизатор Ml узнает на этом этапе впервые, причем данные о ней приходят от двух соседей - от М3 и М4. Поскольку метрики в этих сообщениях указаны одинаковые, то в таблицу попадают данные, которые пришли первыми. В нашем примере считается, что маршрутизатор М2 опередил маршрутизатор М3 и первым переслал свое RIP-сообщение маршрутизатору Ml.

Если маршрутизаторы периодически повторяют этапы рассылки и обработки RIP-сообщений, то за конечное время в сети установится корректный режим маршругизации. Под корректным режимом маршрутизации здесь понимается такое состояние таблиц маршрутизации, когда все сети будут достижимы из любой сети с помощью некоторого рационального маршрута. Пакеты будут доходить до адресатов и не зацикливаться в петлях, подобных той, которая образуется на рис. 5.26, маршрутизаторами М1-М2-М3-М4.

Очевидно, если в сети все маршрутизаторы, их интерфейсы и соединяющие их каналы связи постоянно работоспособны, то объявления по протоколу RIP можно делать достаточно редко, например, один раз в день. Однако в сетях постоянно происходят

изменения - изменяется как работоспособность маршрутизаторов и каналов, так и сами маршрутизаторы и каналы могут добавляться в существующую сеть или же выводиться из ее состава.

Для адаптации к изменениям в сети протокол RIP использует ряд механизмов.

Адаптация RIP-маршрутизаторов к изменениям состояния сети

К новым маршрутам RIP-маршрутизаторы приспосабливаются просто - они передают новую информацию в очередном сообщении своим соседям и постепенно эта информация становится известна всем маршрутизаторам сети. А вот к отрицательным изменениям, связанным с потерей какого-либо маршрута, RIP-маршрутиза-торы приспосабливаются сложнее. Это связано с тем, что в формате сообщений протокола RIP нет поля, которое бы указывало на то, что путь к данной сети больше не существует.

Вместо этого используются два механизма уведомления о том, что некоторый маршрут более недействителен:

- истечение времени жизни маршрута;
- указание специального расстояния (бесконечности) до сети, ставшей недоступной.

Для отработки первого механизма каждая запись таблицы маршрутизации (как и записи таблицы продвижения моста/коммутатора), полученная по протоколу RIP, имеет время жизни (TTL). При поступлении очередного RIP-сообщения, которое подтверждает справедливость данной записи, таймер TTL устанавливается в исходное состояние, а затем из него каждую секунду вычитается единица. Если за время тайм-аута не придет новое маршрутное сообщение об этом маршруте, то он помечается как недействительный.

Время тайм-аута связано с периодом рассылки векторов по сети. В RIP IP период рассылки выбран равным 30 секундам, а в качестве тайм-аута выбрано шестикратное значение периода рассылки, то есть 180 секунд. Выбор достаточно малого времени периода рассылки объясняется несколькими причинами, которые станут понятны из дальнейшего изложения. Шестикратный запас времени нужен для уверенности в том, что сеть действительно стала недоступна, а не просто произошли потери RIP-сообщений (а это возможно, так как RIP использует транспортный протокол UDP, который не обеспечивает надежной доставки сообщений).

Если какой-либо маршрутизатор отказывает и перестает слать своим соседям сообщения о сетях, которые можно достичь через него, то через 180 секунд все записи, которые породил этот маршрутизатор, станут недействительными у его ближайших соседей. После этого процесс повторится уже для соседей ближайших соседей - они вычеркнут подобные записи уже через 360 секунд, так как первые 180 секунд ближайшие соседи еще передавали сообщения об этих записях.

Как видно из объяснения, сведения о недоступных через отказавший маршрутизатор сетях распространяются по сети не очень быстро, время распространения кратно времени жизни записи, а коэффициент кратности равен количеству хопов между самыми дальними маршрутизаторами сети. В этом заключается одна из причин выбора в качестве периода рассылки небольшой величины в 30 секунд.

Если отказывает не маршрутизатор, а интерфейс или сеть, связывающие его с каким-либо соседом, то ситуация сводится к только что описанной - снова начинает работать механизм тайм-аута и ставшие недействительными маршруты постепенно будут вычеркнуты из таблиц всех маршрутизаторов сети.

Тайм-аут работает в тех случаях, когда маршрутизатор не может послать соседям сообщение об отказавшем маршруте, так как либо он сам неработоспособен, либо неработоспособна линия связи, по которой можно было бы передать сообщение.

Когда же сообщение послать можно, RIP-маршрутизаторы не используют специальный признак в сообщении, а указывают бесконечное расстояние до сети, причем в протоколе RIP оно выбрано равным 16 хопам (при другой метрике необходимо указать маршрутизатору ее значение, считающееся бесконечностью). Получив сообщение, в котором некоторая сеть сопровождается расстоянием 16 (или 15, что приводит к тому же результату, так как маршрутизатор наращивает полученное значение на 1), маршрутизатор должен проверить, исходит ли эта «плохая» информация о сети от того же маршрутизатора, сообщение которого послужило в свое время основанием для записи о данной сети в таблице маршрутизации. Если это тот же маршрутизатор, то информация считается достоверной и маршрут помечается как недоступный.

Такое небольшое значение «бесконечного» расстояния вызвано тем, что в некоторых случаях отказы связей в сети вызывают длительные периоды некорректной работы RIP-маршрутизаторов, выражающейся в зацикливании пакетов в петлях сети. И чем меньше расстояние, используемое в качестве «бесконечного», тем такие периоды становятся короче.

Рассмотрим случай зацикливания пакетов на примере сети, изображенной на рис. 5.26.

Пусть маршрутизатор Ml обнаружил, что его связь с непосредственно подключенной сетью 201.36.14.0 потеряна (например, по причине отказа интерфейса 201.36.14.3). Ml отметил в своей таблице маршрутизации, что сеть 201.36.14.0 недоступна. В худшем случае он обнаружил это сразу же после отправки очередных RIP-сообщений, так что до начала нового цикла его объявлений, в котором он должен сообщить соседям, что расстояние до сети 201.36.14.0 стало равным 16, остается почти 30 секунд.

Каждый маршрутизатор работает на основании своего внутреннего таймера, не синхронизируя работу по рассылке объявлений с другими маршрутизаторами. Поэтому весьма вероятно, маршрутизатор М2 опередил маршрутизатор Ml и передал ему свое сообщение раньше, чем Ml успел передать новость о недостижимости сети 201.36.14.0. А в этом сообщении имеются данные, порожденные следующей записью в таблице маршрутизации М2 (табл. 5.18).

Таблица 5.18. Таблица маршрутизации маршрутизатора М2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.7	1	2

Эта запись была получена от маршрутизатора Ml и корректна до отказа интерфейса 201.36.14.3, а теперь она устарела, но маршрутизатор М2 об этом не узнал.

Теперь маршрутизатор Ml получил новую информацию о сети 201.36.14.0 - эта сеть достижима через маршрутизатор М2 с метрикой 2. Раньше Ml также получал эту информацию от М2. Но игнорировал ее, так как его собственная метрика для 201.36.14.0 была лучше. Теперь Ml должен принять данные о сети 201.36.14.0, полученные от М2, и заменить запись в таблице маршрутизации о недостижимости этой сети (табл. 5.19).

Таблица 5.19. Таблица маршрутизации маршрутизатора М1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.101	2	3

В результате в сети образовалась маршрутная петля: пакеты, направляемые узлам сети 201.36.14.0, будут передаваться маршрутизатором М2 маршрутизатору Ml, а маршрутизатор Ml будет возвращать их маршрутизатору М2. IP-пакеты будут циркулировать по этой петле до тех пор, пока не истечет время жизни каждого пакета.

Маршрутная петля будет существовать в сети достаточно долго. Рассмотрим периоды времени, кратные времени жизни записей в таблицах маршрутизаторов.

- Время 0-180 с. После отказа интерфейса в маршрутизаторах Ml и М2 будут сохраняться некорректные записи, приведенные выше. Маршрутизатор М2 по-прежнему снабжает маршрутизатор Ml своей записью о сети 201.36.14.0 с метрикой 2, так как ее время жизни не истекло. Пакеты зацикливаются.
- Время 180-360 с. В начале этого периода у маршрутизатора М2 истекает время жизни записи о сети 201.36.14.0 с метрикой 2, так как маршрутизатор Ml в предыдущий период посылал ему сообщения о сети 201.36.14.0 с худшей метрикой, чем у М2, и они не могли подтверждать эту запись. Теперь маршрутизатор М2 принимает от маршрутизатора Ml запись о сети 201.36.14.0 с метрикой 3 и трансформирует ее в запись с метрикой 4. Маршрутизатор Ml не получает новых сообщений от маршрутизатора М2 о сети 201.36.14.0 с метрикой 2, поэтому время жизни его записи начинает уменьшаться. Пакеты продолжают зацикливаться.
- Время 360-540 с. Теперь у маршрутизатора Ml истекает время жизни записи о сети 201.36.14.0 с метрикой 3. Маршрутизаторы Ml и М2 опять меняются ролями - М2 снабжает Ml устаревшей информацией о пути к сети 201.36.14.0, уже с метрикой 4, которую Ml преобразует в метрику 5. Пакеты продолжают зацикливаться.

Если бы в протоколе RIP не было выбрано расстояние 16 в качестве недостижимого, то описанный процесс длился бы до бесконечности (вернее, пока не была бы исчерпана разрядная сетка поля расстояния и не было бы зафиксировано переполнения при очередном наращивании расстояния).

В результате маршрутизатор М2 на очередном этапе описанного процесса получает от маршрутизатора Ml метрику 15, которая после наращивания, превращаясь в метрику 16, фиксирует недостижимость сети. Период нестабильной работы сети длился 36 минут!

Ограничение в 15 хопов сужает область применения протокола RIP до сетей, в которых число промежуточных маршрутизаторов не может быть больше 15. Для более масштабных сетей нужно применять другие протоколы маршрутизации, например OSPF, или разбивать сеть на автономные области.

Приведенный пример хорошо иллюстрирует главную причину нестабильной работы маршрутизаторов, работающих по протоколу RIP. Эта причина коренится в самом принципе работы дистанционно-векторных протоколов - пользовании информацией, полученной из вторых рук. Действительно, маршрутизатор М2 передал маршрутизатору Ml информацию о достижимости сети 201.36.14.0, за достоверность которой он сам не отвечает. Искоренить эту причину полностью нельзя, ведь сам способ построения таблиц маршрутизации связан с передачей чужой информации без указания источника ее происхождения.

Не следует думать, что при любых отказах интерфейсов и маршрутизаторов в сетях возникают маршрутные петли. Если бы маршрутизатор Ml успел передать сообщение о недостижимости сети 201.36.14.0 раньше ложной информации маршрутизатора М2, то маршрутная петля не образовалась бы. Так что маршрутные петли даже без дополнительных методов борьбы с ними, описанными в следующем разделе, возникают в среднем не более чем в половине потенциально возможных случаев.

Методы борьбы с ложными маршрутами в протоколе RIP

Несмотря на то что протокол RIP не в состоянии полностью исключить переходные состояния в сети, когда некоторые маршрутизаторы пользуются устаревшей информацией об уже несуществующих маршрутах, имеется несколько методов, которые во многих случаях решают подобные проблемы.

Ситуация с петлей, образующейся между соседними маршрутизаторами, описанная в предыдущем разделе, надежно решается с помощью метода, получившем название *расщепления горизонта (split horizon)*. Метод заключается в том, что маршрутная информация о некоторой сети, хранящаяся в таблице маршрутизации, никогда не передается тому маршрутизатору, от которого она получена (это следующий маршрутизатор в данном маршруте). Если маршрутизатор М2 в рассмотренном выше примере поддерживает технику расщепления горизонта, то он не передаст маршрутизатору Ml устаревшую информацию о сети 201.36.14.0, так как получил ее именно от маршрутизатора Ml.

Практически все сегодняшние маршрутизаторы, работающие по протоколу RIP, используют технику расщепления горизонта.

Однако расщепление горизонта не помогает в тех случаях, когда петли образуются не двумя, а несколькими маршрутизаторами. Рассмотрим более детально ситуацию, которая возникнет в сети, приведенной на рис. 5.26, в случае потери связи маршрутизатора 2 с сетью А. Пусть все маршрутизаторы этой сети поддерживают технику расщепления горизонта. Маршрутизаторы М2 и М3 не будут возвращать маршрутизатору в этой ситуации данные о сети 201.36.14.0 с метрикой 2, так как они получили эту информацию

от маршрутизатора Ml. Однако они будут передавать маршрутизатору информацию о достижимости сети 201.36.14.0 с метрикой 4 через себя, так как получили эту информацию по сложному маршруту, а не от маршрутизатора Ml непосредственно. Например, маршрутизатор М2 получил эту информацию по цепочке М4-М3-М1. Поэтому маршрутизатор Ml снова может быть обманут, пока каждый из маршрутизаторов в цепочке М3-М4-М2 не вычеркнет запись о достижимости сети 1 (а это произойдет через период 3 х 180 секунд).

Для предотвращения зацикливания пакетов по составным петлям при отказах связей применяются два других приема, называемые *триггерными обновлениями (triggered updates)* и *замораживанием изменений (hold down).*

Способ триггерных обновлений состоит в том, что маршрутизатор, получив данные об изменении метрики до какой-либо сети, не ждет истечения периода передачи таблицы маршрутизации, а передает данные об изменившемся маршруте немедленно. Этот прием может во многих случаях предотвратить передачу устаревших сведений об отказавшем маршруте, но он перегружает сеть служебными сообщениями, поэтому триггерные объявления также делаются с некоторой задержкой. Поэтому возможна ситуация, когда регулярное обновление в каком-либо маршрутизаторе чуть опередит по времени приход триггерного обновления от предыдущего в цепочке маршрутизатора и данный маршрутизатор успеет передать по сети устаревшую информацию о несуществующем маршруте.

Второй прием позволяет исключить подобные ситуации. Он связан с введением тайм-аута на принятие новых данных о сети, которая только что стала недоступной. Этот тайм-аут предотвращает принятие устаревших сведений о некотором маршруте от тех маршрутизаторов, которые находятся на некотором расстоянии от отказавшей связи и передают устаревшие сведения о ее работоспособности. Предполагается, что в течение тайм-аута «замораживания изменений» эти маршрутизаторы вычеркнут данный маршрут из своих таблиц, так как не получат о нем новых записей и не будут распространять устаревшие сведения по сети.

5.4.3. Протокол «состояния связей» OSPF

Протокол *OSPF (Open Shortest Path First, открытый протокол «кратчайший путь первыми)* является достаточно современной реализацией алгоритма состояния связей (он принят в 1991 году) и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

В OSPF процесс построения таблицы маршрутизации разбивается на два крупных этапа. На первом этапе каждый маршрутизатор строит граф связей сети, в котором вершинами графа являются маршрутизаторы и IP-сети, а ребрами - интерфейсы маршрутизаторов. Все маршрутизаторы для этого обмениваются со своими соседями той информацией о графе сети, которой они располагают к данному моменту времени. Этот процесс похож на процесс распространения векторов расстояний до сетей в протоколе RIP, однако сама информация качественно другая - это информация о топологии сети. Эти сообщения называются *router links advertisement - объявление о связях маршрутизатора*. Кроме того, при передаче топологической информации маршрутизаторы ее не модифицируют, как это делают RIP-маршрутизаторы, а передают в неизменном виде. В результате распространения топологической информации все маршрутизаторы сети располагают

идентичными сведениями о графе сети, которые хранятся в *топологической базе данных* маршрутизатора.

Второй этап состоит в нахождении оптимальных маршрутов с помощью полученного графа. Каждый маршрутизатор считает себя центром сети и ищет оптимальный маршрут до каждой известной ему сети. В каждом найденном таким образом маршруте запоминается только один шаг - до следующего маршрутизатора, в соответствии с принципом одношаговой маршрутизации. Данные об этом шаге и попадают в таблицу маршрутизации. Задача нахождения оптимального пути на графе является достаточно сложной и трудоемкой. В протоколе OSPF для ее решения используется итеративный алгоритм Дийкстры. Если несколько маршрутов имеют одинаковую метрику до сети назначения, то в таблице маршрутизации запоминаются первые шаги всех этих маршрутов.

После первоначального построения таблицы маршрутизации необходимо отслеживать изменения состояния сети и вносить коррективы в таблицу маршрутизации. Для контроля состояния связей и соседних маршрутизаторов OSPF-маршрутиза-торы не используют обмен полной таблицей маршрутизации, как это не очень рационально делают MP-маршрутизаторы. Вместо этого они передают специальные короткие сообщения HELLO. Если состояние сети не меняется, то OSPF-мар-шрутизаторы корректировкой своих таблиц маршрутизации не занимаются и не посылают соседям объявления о связях. Если же состояние связи изменилось, то ближайшим соседям посылается новое объявление, касающееся только данной связи, что, конечно, экономит пропускную способность сети. Получив новое объявление об изменении состояния связи, маршрутизатор перестраивает граф сети, заново ищет оптимальные маршруты (не обязательно все, а только те, на которых отразилось данное изменение) и корректирует свою таблицу маршрутизации. Одновременно маршрутизатор ретранслирует объявление каждому из своих ближайших соседей (кроме того, от которого он получил это объявление).

При появлении новой связи или нового соседа маршрутизатор узнает об этом из новых сообщений HELLO. В сообщениях HELLO указывается достаточно детальная информация о том маршрутизаторе, который послал это сообщение, а также о его ближайших соседях, чтобы данный маршрутизатор можно было однозначно идентифицировать. Сообщения HELLO отправляются через каждые 10 секунд, чтобы повысить скорость адаптации маршрутизаторов к изменениям, происходящим в сети. Небольшой объем этих сообщений делает возможной такое частое тестирование состояния соседей и связей с ними.

Так как маршрутизаторы являются одними из вершин графа, то они обязательно должны иметь идентификаторы.

Протокол OSPF обычно использует метрику, учитывающую пропускную способность сетей. Кроме того, возможно использование двух других метрик, учитывающих требования к качеству обслуживания в IP-пакете, - задержки передачи пакетов и надежности передачи пакетов сетью. Для каждой из метрик протокол OSPF строит отдельную таблицу маршрутизации. Выбор нужной таблицы происходит в зависимости от требований к качеству обслуживания пришедшего пакета (см. рис. 5.27).

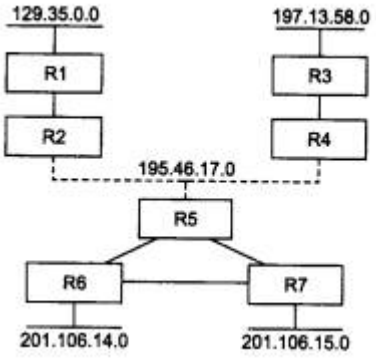


Рис. 5.27. Построение таблицы маршрутизации по протоколу OSPF

Маршрутизаторы соединены как с локальными сетями, так и непосредственно между собой глобальными каналами типа «точка-точка».

Данной сети соответствует граф, приведенный на рис. 5.28.

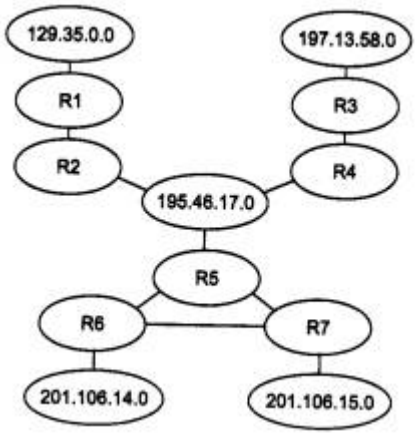


Рис. 5.28. Граф сети, построенный протоколом OSPF

Протокол OSPF в своих объявлениях распространяет информацию о связях двух типов: маршрутизатор - маршрутизатор и маршрутизатор - сеть. Примером связи первого типа служит связь «R3 - R4», а второго - связь «R4 - 195.46.17.0». Если каналам «точка-точка» дать IP-адреса, то они станут дополнительными вершинами графа, как и локальные сети. Вместе с IP-адресом сети передается также информация о маске сети.

После инициализации OSPF-маршрутизаторы знают только о связях с непосредственно подключенными сетями, как и RIP-маршрутизаторы. Они начинают распространять эту информацию своим соседям. Одновременно они посылают сообщения HELLO по всем своим интерфейсам, так что почти сразу же маршрутизатор узнает идентификаторы своих ближайших соседей, что пополняет его топологическую базу новой информацией, которую он узнал непосредственно. Далее топологическая информация начинает распространяться по сети от соседа к соседу и через некоторое время достигает самых удаленных маршрутизаторов.

Каждая связь характеризуется метрикой. Протокол OSPF поддерживает стандартные для многих протоколов (например, для протокола Spanning Tree) значения расстояний для

метрики, отражающей производительность сетей: Ethernet - 10 единиц, Fast Ethernet - 1 единица, канал T1 - 65 единиц, канал 56 Кбит/с - 1785 единиц и т. д.

При выборе оптимального пути на графе с каждым ребром графа связана метрика, которая добавляется к пути, если данное ребро в него входит. Пусть на приведенном примере маршрутизатор R5 связан с .R6 и R7 каналами Tl, а R6 и R7 связаны между собой каналом 56 Кбит/с. Тогда R7 определит оптимальный маршрут до сети 201.106.14.0 как составной, проходящий сначала через маршрутизатор R5, а затем через R6, поскольку у этого маршрута метрика будет равна 65+65 = 130 единиц. Непосредственный маршрут через R6 не будет оптимальным, так как его метрика равна 1785. При использовании хопов был бы выбран маршрут через R6, что не было бы оптимальным.

Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками. Если такие записи образуются в таблице маршрутизации, то маршрутизатор реализует режим баланса загрузки маршрутов (load balancing), отправляя пакеты попеременно по каждому из маршрутов.

У каждой записи в топологической базе данных имеется срок жизни, как и у маршрутных записей протокола RIP. С каждой записью о связях связан таймер, который используется для контроля времени жизни записи. Если какая-либо запись топологической базы маршрутизатора, полученная от другого маршрутизатора, устаревает, то он может запросить ее новую копию с помощью специального сообщения Link-State Request протокола OSPF, на которое должен поступить ответ Link-State Update от маршрутизатора, непосредственно тестирующего запрошенную связь.

При инициализации маршрутизаторов, а также для более надежной синхронизации топологических баз маршрутизаторы периодически обмениваются всеми записями базы, но этот период существенно больше, чем у RIP-маршрутизаторов.

Так как информация о некоторой связи изначально генерируется только тем маршрутизатором, который выяснил фактическое состояние этой связи путем тестирования с помощью сообщений HELLO, а остальные маршрутизаторы только ретранслируют эту информацию без преобразования, то недостоверная информация о достижимости сетей, которая может появляться в RIP-маршрутиэаторах, в OSPF-маршрутизаторах появиться не может, а устаревшая информация быстро заменяется новой, так как при изменении состояния связи новое сообщение генерируется сразу же.

Периоды нестабильной работы в OSPF-сетях могут возникать. Например, при отказе связи, когда информация об этом не дошла до какого-либо маршрутизатора и он отправляет пакеты сети назначения, считая эту связь работоспособной. Однако эти периоды продолжаются недолго, причем пакеты не зацикливаются в маршрутных петлях, а просто отбрасываются при невозможности их передать через неработоспособную связь.

К недостаткам протокола OSPF следует отнести его вычислительную сложность, которая быстро растет с увеличением размерности сети, то есть количества сетей, маршрутизаторов и связей между ними. Для преодоления этого недостатка в протоколе OSPF вводится понятие *области сети (area)* (не нужно путать с автономной системой Internet). Маршрутизаторы, принадлежащие некоторой области, строят граф связей только для этой области, что сокращает размерность сети. Между областями информация о связях не передается, а пограничные для областей маршрутизаторы обмениваются только информацией об адресах сетей, имеющихся в каждой из областей, и расстоянием от

пограничного маршрутизатора до каждой сети. При передаче пакетов между областями выбирается один из пограничных маршрутизаторов области, а именно тот, у которого расстояние до нужной сети меньше. Этот стиль напоминает стиль работы протокола RIP, но нестабильность здесь устраняется тем, что петлевидные связи между областями запрещены. При передаче адресов в другую область OSPF-маршрутизаторы агрегируют несколько адресов в один, если обнаруживают у них общий префикс.

OSPF-маршрутизаторы могут принимать адресную информацию от других протоколов маршрутизации, например от протокола RIP, что полезно для работы в гетерогенных сетях. Такая адресная информация обрабатывается так же, как и внешняя информация между разными областями.

Выводы

- Крупные сети разбивают на автономные системы, в которых проводится общая политика маршрутизации IP-пакетов. Если сеть подключена к Internet, то идентификатор автономной системы назначается в InterNIC.
- Протоколы маршрутизации делятся на внешние и внутренние. Внешние протоколы (EGP, BGP) переносят маршрутную информацию между автономными системами, а внутренние (RIP, OSPF) применяются только в пределах определенной автономной системы.
- Протокол RIP является наиболее заслуженным и распространенным протоколом маршрутизации сетей TCP/IP. Несмотря на его простоту, определенную использованием дистанционно-векторного алгоритма, RIP успешно работает в .небольших сетях с количеством промежуточных маршрутизаторов не более 15.
- RIP-маршрутизаторы при выборе маршрута обычно используют самую простую метрику - количество промежуточных маршрутизаторов между сетями, то есть хопов.
- Версия RIPvl не распространяет маски подсетей, что вынуждает администраторов использовать маски фиксированной длины во всей составной сети. В версии RIPv2 это ограничение снято.
- В сетях, использующих RIP и имеющих петлевидные маршруты, могут наблюдаться достаточно длительные периоды нестабильной работы, когда пакеты зацикливаются в маршрутных петлях и не доходят до адресатов. Для борьбы с этими явлениями в RIP-маршрутизаторах предусмотрено несколько приемов (Split Horizon, Hold Down, Triggered Updates), которые сокращают в некоторых случаях периоды нестабильности.
- Протокол OSPF был разработан для эффективной маршрутизации IP-пакетов в больших сетях со сложной топологией, включающей петли. Он основан на алгоритме состояния связей, который обладает высокой устойчивостью к изменениям топологии сети.
- При выборе маршрута OSPF-маршрутизаторы используют метрику, учитывающую пропускную способность составных сетей.
- Протокол OSPF является первым протоколом маршрутизации для IP-сетей, который учитывает биты качества обслуживания (пропускная способность, задержка и надежность) в заголовке IP-пакета. Для каждого типа качества обслуживания строится отдельная таблица маршрутизации.
- Протокол OSPF обладает высокой вычислительной сложностью, поэтому чаще всего работает на мощных аппаратных маршрутизаторах.

5.5. Средства построения составных сетей стека Novell

5.5.1. Общая характеристика протокола IPX

Протокол *Internetwork Packet Exchange (IPX)* является оригинальным протоколом сетевого уровня стека Novell, разработанным в начале 80-х годов на основе протокола Internetwork Datagram Protocol (IDP) компании Xerox.

Протокол IPX соответствует сетевому уровню модели ISO/OSI (рис. 5.29) и поддерживает, как и протокол IP, только дейтаграммный (без установления соединений) способ обмена сообщениями. В сети NetWare наиболее быстрая передача данных при наиболее экономном использовании памяти реализуется именно протоколом IPX.

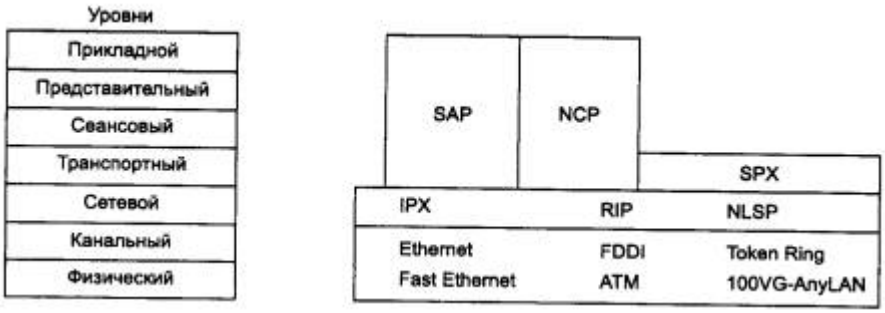


Рис. 5.29. Соответствие протоколов IPX/SPX семиуровневой модели OSI

Надежную передачу пакетов может осуществлять транспортный протокол SPX (Sequenced Packet Exchange Protocol), который работает с установлением соединения и восстанавливает пакеты при их потере или повреждении. Как видно из рис. 5.29, использование протокола SPX не является обязательным при выполнении операций передачи сообщений протоколами прикладного уровня.

Прикладной уровень стека IPX/SPX составляют два протокола: NCP и SAP. Протокол NCP (NetWare Core Protocol) поддерживает все основные службы операционной системы Novell NetWare - файловую службу, службу печати и т. д. Протокол SAP (Service Advertising Protocol) выполняет вспомогательную роль. С помощью протокола SAP каждый компьютер, который готов предоставить какую-либо службу для клиентов сети, объявляет об этом широковещательно по сети, указывая в SAP-пакетах тип службы (например, файловая), а также свой сетевой адрес. Наличие протокола SAP позволяет резко уменьшить административные работы по конфигурированию клиентского программного обеспечения, так как всю необходимую информацию для работы клиенты узнают из объявлений SAP (кроме маршрутизаторов по умолчанию, о которых можно узнать с помощью протокола IPX).

В отличие от протокола IP, который изначально разрабатывался для глобальных сетей, протокол IPX создавался для применения в локальных сетях. Именно поэтому он является одним из самых экономичных протоколов в отношении требований к вычислительным ресурсам и хорошо работает в сравнительно небольших локальных сетях.

Специфика адресации в протоколе IPX является источником как достоинств, так и недостатков этого протокола. Протокол IPX работает с сетевыми адресами, включающими три компонента:

- номер сети (4 байта);
- номер узла (6 байт);
- номер сокета (2 байта).

Номер сети в отличие от протокола IP имеет всегда фиксированную длину - 4 байта. В принципе для корпоративных сетей эта длина является избыточной, так как вряд ли у предприятия возникнет потребность разделить свою сеть на 4 миллиарда подсетей. В период доминирования сетей IPX/SPX компания Novell рассматривала возможность создания единого всемирного центра по распределению IPX-адресов, аналогичного центру InterNIC. Однако стремительный рост популярности сети Internet лишил это начинание смысла. Хотя протоколы IPX/SPX по-прежнему работают в огромном количестве корпоративных сетей, заменить IP во всемирной сети они уже не смогут. Надо отметить, что специалисты компании Novell приложили немало усилий, чтобы в новой версии 6 протокол IP приобрел некоторые черты, свойственные протоколу IPX, и тем самым облегчил переход пользователей IPX на IPv6 (когда это станет практически необходимым). Обычно все три составляющие IPX-адреса, в том числе и номер сети, записываются в шест-надцатеричной форме.

Под номером узла в протоколе IPX понимается аппаратный адрес узла. В локальных сетях это MAC - адрес узла - сетевого адаптера или порта маршрутизатора. Размер адреса узла в 6 байт отражает происхождение этого поля, но в него можно поместить любой аппаратный адрес, если он укладывается в размер этого поля.

Номер сокета (socket) идентифицирует приложение, которое передает свои сообщения по протоколу IPX. Сокет выполняет в стеке IPX/SPX ту же роль, что порт в протоколах TCP/UDP стека TCP/IP. Наличие этого поля в протоколе сетевого уровня, которым является IPX, объясняется тем, что в стеке Novell прикладные протоколы NCP и SAP взаимодействует с сетевым уровнем непосредственно, минуя транспортный протокол SPX. Поэтому роль мультиплексора-демультиплексора прикладных протоколов приходится выполнять протоколу IPX, для чего в его пакете необходимо передавать номер сокета прикладного протокола. Протоколы NCP и SAP не пользуются услугами SPX для ускорения работы стека, а скорость работы на маломощных персональных компьютерах начала 80-х годов была одной из основных целей компании Novell. Каждый дополнительный уровень в стеке, хотя бы и такой простой, как UDP, замедляет работу стека. За отказ от транспортного уровня компании Novell пришлось реализовывать средства восстановления утерянных пакетов в протоколе NCP. Тем не менее прикладные программисты, разрабатывающие свои собственные сетевые приложения для стека IPX/SPX, могут пользоваться протоколом SPX, если не захотят встраивать достаточно сложные алгоритмы скользящего окна в свои программы.

Протокол IPX является одним из наиболее легко настраиваемых протоколов сетевого уровня. Номер сети задается администратором только на серверах, а номер узла автоматически считывается из сетевого адаптера компьютера. На клиентском компьютере номер сети не задается - клиент узнает эту информацию из серверных объявлений SAP или локального маршрутизатора.

Адрес маршрутизатора по умолчанию также не нужно задавать вручную на каждом клиентском компьютере. В протоколе IPX есть специальный запрос, который передается на заранее определенный номер сокета. Если в сети клиента есть маршрутизатор или сервер, выполняющий роль программного маршрутизатора, то клиент при старте системы

выдает такой запрос широковещательно, и все маршрутизаторы сообщают ему свои MAC - адреса, которые используются в качестве адреса следующего маршрутизатора.

Как видно из описания, административные издержки при конфигурировании сети IPX/SPX сводятся к минимуму. При этом отпадает необходимость в протоколе типа ARP, выясняющего соответствие между сетевыми адресами узлов и их MAC - адресами. Однако при смене сетевого адаптера нужно скорректировать адрес узла, если для его выяснения используются не широковещательные запросы-ответы, а справочная служба типа Novell NDS, в которой фиксируются сетевые адреса серверов. Отсутствие протокола ARP повышает производительность сети, так как позволяет не тратить время на выполнение ARP-запросов и ARP-ответов.

5.5.2. Формат пакета протокола IPX

Пакет протокола IPX имеет гораздо более простую структуру по сравнению с пакетом IP, что, собственно, и отражает меньшие функциональные возможности протокола IPX.

IPX-пакет имеет следующие поля.

- *Контрольная сумма (Checksum)* - это 2-байтовое поле, являющееся «пережитком прошлого», которое протокол IPX ведет от протокола IDP стека Xerox. Так как низкоуровневые протоколы (например, Ethernet) всегда выполняют проверку контрольных сумм, то IPX не использует это поле и всегда устанавливает его в единицы.
- *Длина (Length)* занимает 2 байта и задает размер всего пакета, включая IPX-заголовок и поле данных. Самый короткий пакет - 30 байт - включает только IPX-заголовок, а рекомендуемый максимально большой - 576 байт - включает IPX-заголовок плюс 546 байт данных. Максимальный размер пакета в 576 байт соответствует рекомендациям стандартов Internet для составных сетей. Протокол IPX вычисляет значение этого поля, основываясь на информации, предоставляемой прикладной программой при вызове функции IPX. IPX-пакет может превосходить рекомендуемый максимум в 576 байт, что и происходит в локальных сетях Ethernet, где используются IPX-пакеты в 1500 байт с полем данных в 1470 байт.
- *Управление транспортом (Transport control)* имеет длину 8 бит. Это поле определяет время жизни пакета в хопах. IPX-пакет может пересечь до 15 маршрутизаторов. Протокол IPX устанавливает это однобайтовое поле в 0 до начала передачи, а затем увеличивает его на 1 каждый раз, когда пакет проходит через маршрутизатор. Если счетчик превысит 15, то пакет аннулируется.
- *Тип пакета (Packet type)* имеет длину 8 бит. Фирма Xerox определила в свое время определенные значения для различных типов пакетов: прикладные программы, посылающие IPX-пакеты, должны устанавливать это поле в значение, равное 4. Значение 5 соответствует служебным IPX-пакетам, используемым протоколом SPX в качестве служебных сообщений. Значение 17 указывает на то, что в поле данных IPX-пакета находится сообщение протокола NetWare Core Protocol (NCP) - основного протокола файловой службы NetWare.
- *Адрес назначения (Destination address)* - состоит из трех полей: номера сети назначения, номера узла назначения, номера сокета назначения. Эти поля занимают соответственно 4, 6 и 2 байта.
- *Адрес отправителя (Source address)* - номер исходной сети, номер исходного узла, номер исходного сокета. Аналогичны адресным полям назначения.

- *Поле данных (Data)*. Может занимать от 0 до 546 байт. Поле данных нулевой длины может использоваться в служебных пакетах, например, для подтверждения получения предыдущего пакета. Из анализа формата пакета можно сделать некоторые выводы об ограничениях протокола IPX.
- *Отсутствует возможность динамической фрагментации на сетевом уровне*. В IPX-пакете нет полей, с помощью которых маршрутизатор может разбить слишком большой пакет на части. При передаче пакета в сеть с меньшим значением MTU IPX-маршрутизатор отбрасывает пакет. Протокол верхнего уровня, например NCP, должен последовательно уменьшать размер пакета до тех пор, пока не получит на него положительную квитанцию.
- *Большие накладные расходы на служебную информацию*. Сравнительно небольшая максимальная длина поля данных IPX-пакета (546 байт при длине заголовка 30 байт) приводит к тому, что как минимум 5 % данных являются служебными.
- *Время жизни пакета ограничено числом 15*, что может оказаться недостаточным для большой сети (для сравнения, в IP-сетях пакет может пройти до 255 промежуточных маршрутизаторов).
- *Отсутствует поле качества сервиса*, что не позволяет маршрутизаторам автоматически подстраиваться к требованиям приложения к качеству передачи трафика.

Кроме того, некоторые недостатки сетей Novell связаны не с протоколом IPX, а со свойствами других протоколов стека IPX/SPX. Многие недостатки проявляются при работе стека IPX/SPX на медленных глобальных линиях связи, и это закономерно, так как ОС NetWare оптимизировалась для работы в локальной сети.

Например, неэффективная работа по восстановлению потерянных и искаженных пакетов на низкоскоростных глобальных каналах обусловлена тем, что протокол NCP, который выполняет эту работу, использует метод получения квитанций с простоями. В локальных сетях со скоростью 10 Мбит/с такой метод работал вполне эффективно, а на медленных каналах время ожидания квитанции заметно тормозит работу передающего узла.

В версиях ОС NetWare до 4.0 соответствие символьных имен серверов их сетевым адресам устанавливалось только с помощью широковещательного протокола Service Advertising Protocol (SAP). Однако широковещательные рассылки заметно засоряют медленные глобальные каналы. Модернизируя свой стек для применения в крупных корпоративных сетях, компания Novell использует теперь справочную службу NDS (NetWare Directory Services) для нахождения разнообразной информации об имеющихся в сети ресурсах и службах, в том числе и о соответствии имени сервера его сетевому адресу. Так как служба NDS поддерживается только серверами с версией NetWare 4.x и выше, то для работы с версиями NetWare 3.x маршрутизаторы распознают SAP-пакеты по номеру их сокета и передают их на все порты, имитируя широковещательные рассылки локальной сети, на что тратится значительная часть пропускной способности медленных глобальных линий. Кроме того, такая «псевдошироковещательность» сводит на нет изоляцию сетей от некорректных SAP-пакетов.

В последних версиях своей операционной системы NetWare компания Novell значительно модифицировала свой стек для того, чтобы он мог более эффективно использоваться в крупных составных сетях.

- Служба NDS позволяет отказаться от широковещательного протокола SAP. Служба NDS основана на иерархической распределенной базе данных, хранящей

информацию о пользователях и разделяемых ресурсах сети. Приложения обращаются к этой службе по протоколу прикладного уровня NDS.
- Добавлен модуль для реализации метода скользящего окна - так называемый Burst Mode Protocol NLM.
- Добавлен модуль для поддержки длинных IPX-пакетов в глобальных сетях - Large Internet Packet NLM.

Кроме того, постоянное повышение быстродействия глобальных служб уменьшает недостатки оригинальных протоколов стека IPX/SPX, что позволяет некоторым обозревателям говорить об успешной работе операционной системы NetWare в глобальных сетях и без указанных нововведений.

5.5.3. Маршрутизация протокола IPX

В целом маршрутизация протокола IPX выполняется аналогично маршрутизации протокола IP. Каждый IPX-маршрутизатор поддерживает таблицу маршрутизации, на основании которой принимается решение о продвижении пакета. IPX-маршрутизаторы поддерживает одношаговую маршрутизацию, при которой каждый маршрутизатор принимает решение только о выборе следующего на пути маршрутизатора. Возможности маршрутизации от источника в протоколе IPX отсутствуют. Рассмотрим типичную таблицу маршрутизации (табл. 5.20) для протокола IPX.

Таблица 5.20. Таблица маршрутизации протокола IPX

Номер сети	Следующий маршрутизатор	Порт	Задержка	Хопы
A0000010	—	1	0	0
A0000011	—	2	0	0
000013F4	A0000010-008100E30067	1	3	2
00000120	A0000011-C000023300FA	2	2	1
00000033	A0000010-008100E30055	1	10	5

В поле «Номер сети» указывается шестнадцатеричный адрес сети назначения, а в поле «Следующий маршрутизатор» - полный сетевой адрес следующего маршрутизатора, то есть пара «номер сети-MAC - адрес». MAC - адрес из этой записи переносится в поле адреса назначения кадра канального уровня, например Ethernet, который и переносит IPX-пакет следующему маршрутизатору. IPX-пакет при передаче между промежуточными маршрутизаторами изменений не претерпевает.

Если IPX-маршрутизатор обнаруживает, что сеть назначения - это его непосредственно подключенная сеть, то из заголовка IPX-пакета извлекается номер узла назначения, который является MAC - адресом узла назначения. Этот MAC - адрес переносится в адрес назначения кадра канального уровня, например FDDI. Кадр непосредственно отправляется в сеть, и протокол FDDI доставляет его по этому адресу узлу назначения.

IPX-маршрутизаторы обычно используют два типа метрики при выборе маршрута: расстояние в хопах и задержку в некоторых условных единицах - тиках (ticks). Расстояние в хопах имеет обычный смысл - это количество промежуточных маршрутизаторов,

которые нужно пересечь IPX-пакету для достижения сети назначения. Задержка также часто используется в маршрутизаторах и мостах/коммутаторах для более точного сравнения маршрутов. Однако в IPX-маршрутизаторах традиционно задержка измеряется в тиках таймера персонального компьютера, который выдает сигнал прерывания 18,21 раза в секунду. Эта традиция ведется от первых программных IPX-маршрутизаторов, которые работали в составе операционной системы NetWare и пользовались таймером персонального компьютера для измерения интервалов времени. Напомним, что IP-маршрутизаторы, а также мосты/коммутаторы, поддерживающие протокол Spanning Tree, измеряют задержку, вносимую какой-либо сетью в 10-наносекундных единицах передачи одного бита информации, так что сеть Ethernet оценивается задержкой в 10 единиц. Кроме этого, IPX-маршрутизаторы оценивают задержку не одного бита, а стандартного для IPX-пакета в 576 байт.

Поэтому задержка в тиках для сети Ethernet получается равной 0,00839 тика, а для канала 64 Кбит/с - 1,31 тика. Задержка в тиках всегда округляется до целого числа тиков в большую сторону, так что сеть Ethernet вносит задержку в один тик, а канал 64 Кбит/с - в 2 тика. При вычислении метрики в тиках для составного маршрута задержки в тиках складываются.

Две метрики в записях таблицы маршрутизации протокола IPX используются в порядке приоритетов. Наибольшим приоритетом обладает метрика, измеренная в задержках, а если эта метрика совпадает для каких-либо маршрутов, то во внимание принимается расстояние в хопах.

Несмотря на традиции измерения задержки в тиках, IPX-маршрутизаторы могут использовать и стандартные задержки сетей, измеренные в 10-наносекундных интервалах.

IPX-маршрутизаторы могут поддерживать как статические маршруты, так и динамические, полученные с помощью протоколов RIP IPX и NLSP.

Протокол RIP IPX очень близок к протоколу RIP IP. Так как в IPX-сетях маски не применяются, то RIP IPX не имеет аналога RIPv2, передающего маски. Интервал между объявлениями у протокола RIP IPX равен 60 с (в отличие от 30 су RIP IP). В пакетах RIP IPX для каждой сети указываются обе метрики - в хопах и тиках. Для исключения маршрутных петель IPX-маршрутизаторы используют прием расщепления горизонта.

Время жизни динамической записи составляет 180 секунд. Недостижимость сети указывается значением числа хопов в 15 (0xF), а тиков - в 0xFFFF.

IPX-маршрутизаторы, как и IP-маршрутизаторы, не передают из сети в сеть пакеты, имеющие широковещательный сетевой адрес. Однако для некоторых типов таких пакетов IPX-маршрутизаторы делают исключения. Это пакеты службы SAP, с помощью которой серверы NetWare объявляют о себе по сети. IPX-маршрутизаторы передают SAP-пакеты во все непосредственно подключенные сети, кроме той, от которой этот пакет получен (расщепление горизонта). Если бы IPX-маршрутизаторы не выполняли таких передач, то клиенты NetWare не смогли бы взаимодействовать с серверами в сети, разделенной маршрутизаторами, в привычном стиле, то есть путем просмотра имеющихся серверов с помощью команды SLIST.

IPX-маршрутизаторы всегда используют внутренний номер сети, который относится не к интерфейсам маршрутизатора, а к самому модулю маршрутизации. Внутренний номер

сети является некоторым аналогом сети 127.0.0.0 узлов IP-сетей, однако каждый IPX-маршрутизатор должен иметь уникальный внутренний номер сети, причем его уникальность должна распространяться и на внешние номера IPX-сетей в составной сети.

IPX-маршрутизаторы выполняют также функцию согласования форматов кадров Ethernet. В составных IPX-сетях каждая сеть может работать только с одним из 4-х возможных типов кадров IPX. Поэтому если в разных сетях используются разные типы кадров Ethernet, то маршрутизатор посылает в каждую сеть тот тип кадра, который установлен для этой сети.

Протокол NLSP (NetWare Link Services Protocol) представляет собой реализацию алгоритма состояния связей для IPX-сетей. В основном он работает аналогично протоколу OSPF сетей TCP/IP.

Выводы

- Стек Novell состоит из четырех уровней: канального, который собственно стеком Novell не определяется; сетевого, представленного протоколом дейтаграмм-ного типа IPX; транспортного, на котором работает протокол надежной передачи данных SPX; прикладного, на котором работает протокол NCP, поддерживающий файловую службу и службу печати, а также протоколы SAP и NDS, выполняющие служебные функции по поиску в сети разделяемых ресурсов.
- Особенностью стека Novell является то, что основной прикладной протокол NCP не пользуется транспортным протоколом SPX, а обращается непосредственно к сетевому протоколу IPX. Это значительно ускоряет работу стека, но усложняет прикладной протокол NCP.
- Сетевой IPX-адрес состоит из номера сети, назначаемого администратором, и номера узла, который в локальных сетях совпадает с аппаратным адресом узла, то есть MAC - адресом. Использование аппаратных адресов узлов на сетевом уровне ускоряет работу протокола, так как при этом отпадает необходимость в выполнении протокола типа ARP. Также упрощается конфигурирование компьютеров сети, так как они узнают свой номер сети от локального маршрутизатора, а номер узла извлекается из сетевого адаптера.
- Недостатком IPX-адресации является ограничение в 6 байт, накладываемое на адрес узла на сетевом уровне. Если какая-либо составная сеть использует аппаратные адреса большего размера (это может произойти, например, в сети X.25), то протокол IPX не сможет доставить пакет конечному узлу такой сети.
- IPX-маршрутизаторы используют протоколы динамической маршрутизации RIP IPX, являющийся аналогом RIP IP, и NLSP, который во многом похож на протокол OSPF сетей TCP/IP.

5.6. Основные характеристики маршрутизаторов и концентраторов

5.6.1. Маршрутизаторы

Основная задача маршрутизатора - выбор наилучшего маршрута в сети - часто является достаточно сложной с математической точки зрения. Особенно интенсивных вычислений требуют протоколы, основанные на алгоритме состояния связей, вычисляющие оптимальный путь на графе, - OSPF, NLSP, IS-IS. Кроме этой основной функции в круг

ответственности маршрутизатора входят и другие задачи, такие как буферизация, фильтрация и фрагментация перемещаемых пакетов. При этом очень важна производительность, с которой маршрутизатор выполняет эти задачи.

Поэтому типичный маршрутизатор является мощным вычислительным устройством с одним или даже несколькими процессорами, часто специализированными или построенными на RISC-архитектуре, со сложным программным обеспечением. То есть сегодняшний маршрутизатор - это специализированный компьютер, имеющий скоростную внутреннюю шину или шины (с пропускной способностью 600-2000 Мбит/с), часто использующий симметричное или асимметричное мультипроцессирование и работающий под управлением специализированной операционной системы, относящейся к классу систем реального времени. Многие разработчики маршрутизаторов построили в свое время такие операционные системы на базе операционной системы Unix, естественно, значительно ее переработав.

Маршрутизаторы могут поддерживать как один протокол сетевого уровня (например, IP, IPX или DECnet), так и множество таких протоколов. В последнем случае они называются *многопротокольными* маршрутизаторами. Чем больше протоколов сетевого уровня поддерживает маршрутизатор, тем лучше он подходит для корпоративной сети.

Большая вычислительная мощность позволяет маршрутизаторам наряду с основной работой по выбору оптимального маршрута выполнять и ряд вспомогательных высокоуровневых функций.

Классификация маршрутизаторов по областям применения

По областям применения маршрутизаторы делятся на несколько классов.

Магистральные маршрутизаторы (backbone routers) предназначены для построения центральной сети корпорации. Центральная сеть может состоять из большого количества локальных сетей, разбросанных по разным зданиям и использующих самые разнообразные сетевые технологии, типы компьютеров и операционных систем. Магистральные маршрутизаторы - это наиболее мощные устройства, способные обрабатывать несколько сотен тысяч или даже несколько миллионов пакетов в секунду, имеющие большое количество интерфейсов локальных и глобальных сетей. Поддерживаются не только среднескоростные интерфейсы глобальных сетей, такие как T1/E1, но и высокоскоростные, например, ATM или SDH со скоростями 155 Мбит/с или 622 Мбит/с. Чаще всего магистральный маршрутизатор конструктивно выполнен по модульной схеме на основе шасси с большим количеством слотов - до 12-14. Большое внимание уделяется в магистральных моделях надежности и отказоустойчивости маршрутизатора, которая достигается за счет системы терморегуляции, избыточных источников питания, заменяемых «на ходу» (hot swap) модулей, а также симметричного муль-типроцессирования. Примерами магистральных маршрутизаторов могут служить маршрутизаторы Backbone Concentrator Node (BCN) компании Nortel Networks (ранее Bay Networks), Cisco 7500, Cisco 12000.

Маршрутизаторы региональных отделений соединяют региональные отделения между собой и с центральной сетью. Сеть регионального отделения, так же как и центральная сеть, может состоять из нескольких локальных сетей. Такой маршрутизатор обычно представляет собой некоторую упрощенную версию магистрального маршрутизатора. Если он выполнен на основе шасси, то количество слотов его шасси меньше: 4-5.

Возможен также конструктив с фиксированным количеством портов. Поддерживаемые интерфейсы локальных и глобальных сетей менее скоростные. Примерами маршрутизаторов региональных отделений могут служить маршрутизаторы BLN, ASN компании Nortel Networks, Cisco 3600, Cisco 2500, NetBuilder II компании 3Com. Это наиболее обширный класс выпускаемых маршрутизаторов, характеристики которых могут приближаться к характеристикам магистральных маршрутизаторов, а могут и опускаться до характеристик маршрутизаторов удаленных офисов.

Маршрутизаторы удаленных офисов соединяют, как правило, единственную локальную сеть удаленного офиса с центральной сетью или сетью регионального отделения по глобальной связи. В максимальном варианте такие маршрутизаторы могут поддерживать и два интерфейса локальных сетей. Как правило, интерфейс локальной сети - это Ethernet 10 Мбит/с, а интерфейс глобальной сети - выделенная линия со скоростью 64 Кбит/с, 1,544 или 2 Мбит/с. Маршрутизатор удаленного офиса может поддерживать работу по коммутируемой телефонной линии в качестве резервной связи для выделенного канала. Существует очень большое количество типов маршрутизаторов удаленных офисов. Это объясняется как массовостью потенциальных потребителей, так и специализацией такого типа устройств, проявляющейся в поддержке одного конкретного типа глобальной связи. Например, существуют маршрутизаторы, работающие только по сети ISDN, существуют модели только для аналоговых выделенных линий и т. п. Типичными представителями этого класса являются маршрутизаторы Nautika компании Nortel Networks, Cisco 1600, Office Connect компании 3Com, семейство Pipeline компании Ascend.

Маршрутизаторы локальных сетей (коммутаторы 3-го уровня) предназначены для разделения крупных локальных сетей на подсети. Основное требование, предъявляемое к ним, - высокая скорость маршрутизации, так как в такой конфигурации отсутствуют низкоскоростные порты, такие как модемные порты 33,6 Кбит/с или цифровые порты 64 Кбит/с. Все порты имеют скорость по крайней мере 10 Мбит/с, а многие работают на скорости 100 Мбит/с. Примерами коммутаторов 3-го уровня служат коммутаторы CoreBuilder 3500 компании 3Com, Accelar 1200 компании Nortel Networks, Waveswitch 9000 компании Plaintree, Turboiron Switching Router компании Foudry Networks.

В зависимости от области применения маршрутизаторы обладают различными основными и дополнительными техническими характеристиками.

Основные технические характеристики маршрутизатора

Основные технические характеристики маршрутизатора связаны с тем, как он решает свою главную задачу - маршрутизацию пакетов в составной сети. Именно эти характеристики прежде всего определяют возможности и сферу применения того или иного маршрутизатора.

Перечень поддерживаемых сетевых протоколов. Магистральный маршрутизатор должен поддерживать большое количество сетевых протоколов и протоколов маршрутизации, чтобы обеспечивать трафик всех существующих на предприятии вычислительных систем (в том числе и устаревших, но все еще успешно эксплуатирующихся, так называемых унаследованных - legacy), а также систем, которые могут появиться на предприятии в ближайшем будущем. Если центральная сеть образует отдельную автономную систему Internet, то потребуется поддержка и специфических протоколов маршрутизации этой сети, таких как EGP и BGP. Программное обеспечение магистральных маршрутизаторов обычно строится по модульному принципу, поэтому при возникновении потребности

можно докупать и добавлять программные модули, реализующие недостающие протоколы.

Перечень поддерживаемых сетевых протоколов обычно включает протоколы IP, CONS и CLNS OSI, IPX, AppleTalk, DECnet, Banyan VINES, Xerox XNS.

Перечень протоколов маршрутизации составляют протоколы IP RIP, IPX RIP, NLSP, OSPF, IS-IS OSI, EGP, BGP, VINES RTP, AppleTalk RTMP.

Перечень поддерживаемых интерфейсов локальных и глобальных сетей. Для локальных сетей - это интерфейсы, реализующие физические и канальные протоколы сетей Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN иATM.

Для глобальных связей - это интерфейсы физического уровня для связи с аппаратурой передачи данных, а также протоколы канального и сетевого уровней, необходимые для подключения к глобальным сетям с коммутацией каналов и пакетов.

Поддерживаются интерфейсы последовательных линий (serial lines) RS-232, RS-449/422, V.35 (для передачи данных со скоростями до 2-6 Мбит/с), высокоскоростной интерфейс HSSI, обеспечивающий скорость до 52 Мбит/с, а также интерфейсы с цифровыми каналами T1/E1, T3/E3 и интерфейсами BRI и PRI цифровой сети ISDN. Некоторые маршрутизаторы имеют аппаратуру связи с цифровыми глобальными каналами, что исключает необходимость использования внешних устройств сопряжения с этими каналами.

В набор поддерживаемых глобальных технологий обычно входят технологии X.25, frame relay, ISDN и коммутируемых аналоговых телефонных сетей, сетей ATM, а также поддержка протокола канального уровня PPP.

Общая производительность маршрутизатора. Высокая производительность маршрутизации важна для работы с высокоскоростными локальными сетями, а также для поддержки новых высокоскоростных глобальных технологий, таких как frame relay, T3/E3, SDH и ATM. Общая производительность маршрутизатора зависит от многих факторов, наиболее важными из которых являются: тип используемых процессоров, эффективность программной реализации протоколов, архитектурная организация вычислительных и интерфейсных модулей. Общая производительность маршрутизаторов колеблется от нескольких десятков тысяч пакетов в секунду до нескольких миллионов пакетов в секунду. Наиболее производительные маршрутизаторы имеют мультипроцессорную архитектуру, сочетающую симметричные и асимметричные свойства - несколько мощных центральных процессоров по симметричной схеме выполняют функции вычисления таблицы маршрутизации, а менее мощные процессоры в интерфейсных модулях занимаются передачей пакетов на подключенные к ним сети и пересылкой пакетов на основании части таблицы маршрутизации, кэшированной в локальной памяти интерфейсного модуля.

Магистральные маршрутизаторы обычно поддерживают максимальный набор протоколов и интерфейсов и обладают высокой общей производительностью в один-два миллиона пакетов в секунду. Маршрутизаторы удаленных офисов поддерживают один-два протокола локальных сетей и низкоскоростные глобальные протоколы, общая производительность таких маршрутизаторов обычно составляет от 5 до 20-30 тысяч пакетов в секунду.

Маршрутизаторы региональных отделений занимают промежуточное положение, поэтому их иногда не выделяют в отдельный класс устройств.

Наиболее высокой производительностью обладают коммутаторы 3-го уровня, особенности которых рассмотрены ниже.

Дополнительные функциональные возможности маршрутизаторов

Наряду с функцией маршрутизации многие маршрутизаторы обладают следующими важными дополнительными функциональными возможностями, которые значительно расширяют сферу применения этих устройств.

Поддержка одновременно нескольких протоколов маршрутизации. В протоколах маршрутизации обычно предполагается, что маршрутизатор строит свою таблицу на основе работы только этого одного протокола. Деление Internet на автономные системы также направлено на исключение использования в одной автономной системе нескольких протоколов маршрутизации. Тем не менее иногда в большой корпоративной сети приходится поддерживать одновременно несколько таких протоколов, чаще всего это складывается исторически. При этом таблица маршрутизации может получаться противоречивой - разные протоколы маршрутизации могут выбрать разные следующие маршрутизаторы для какой-либо сети назначения. Большинство маршрутизаторов решает эту проблему за счет придания приоритетов решениям разных протоколов маршрутизации. Высший приоритет отдается статическим маршрутам (администратор всегда прав), следующий приоритет имеют маршруты, выбранные протоколами состояния связей, такими как OSPF или NLSP, а низшим приоритетов обладают маршруты дистанционно-векторных протоколов, как самых несовершенных.

Приоритеты сетевых протоколов. Можно установить приоритет одного протокола сетевого уровня над другими. На выбор маршрутов эти приоритеты не оказывают никакого влияния, они влияют только на порядок, в котором многопротокольный маршрутизатор обслуживает пакеты разных сетевых протоколов. Это свойство бывает полезно в случае недостаточной полосы пропускания кабельной системы и существования трафика, чувствительного к временным задержкам, напримертрафика SNA или голосового трафика, передаваемого одним из сетевых протоколов.

Поддержка политики маршрутных объявлений. В большинстве протоколов обмена маршрутной информации (RIP, OSPF, NLSP) предполагается, что маршрутизатор объявляет в своих сообщениях обо всех сетях, которые ему известны. Аналогично предполагается, что маршрутизатор при построении своей таблицы учитывает все адреса сетей, которые поступают ему от других маршрутизаторов сети. Однако существуют ситуации, когда администратор хотел бы скрыть существование некоторых сетей в определенной части своей сети от других администраторов, например, по соображениям безопасности. Или же администратор хотел бы запретить некоторые маршруты, которые могли бы существовать в сети. При статическом построении таблиц маршрутизации решение таких проблем не составляет труда. Динамические же протоколы маршрутизации не позволяют стандартным способом реализовывать подобные ограничения. Существует только один широко используемый протокол динамической маршрутизации, в котором описана возможность существования *правил (policy)*, ограничивающих распространение некоторых адресов в объявлениях, - это протокол BGP. Необходимость поддержки таких правил в протоколе BGP понятна, так как это протокол обмена маршрутной информацией между автономными системами, где велика потребность в административном

регулировании маршрутов (например, некоторый поставщик услуг Internet может не захотеть, чтобы через него транзитом проходил трафик другого поставщика услуг). Разработчики маршрутизаторов исправляют этот недостаток стандартов протоколов, вводя в маршрутизаторы поддержку правил передачи и использования маршрутной информации, подобных тем, которые рекомендует BGP.

Защита от широковещательных штормов (broadcast storm). Одна из характерных неисправностей сетевого программного обеспечения - самопроизвольная генерация с высокой интенсивностью широковещательных пакетов. Широковещательным штормом считается ситуация, в которой процент широковещательных пакетов превышает 20 % от общего количества пакетов в сети. Обычный коммутатор или мост слепо передает такие пакеты на все свои порты, как того требует его логика работы, засоряя, таким образом, сеть. Борьба с широковещательным штормом в сети, соединенной коммутаторами, требует от администратора отключения портов, генерирующих широковещательные пакеты. Маршрутизатор не распространяет такие поврежденные пакеты, поскольку в круг его задач не входит копирование широковещательных пакетов во все объединяемые им сети. Поэтому маршрутизатор является прекрасным средством борьбы с широковещательным штормом, правда, если сеть разделена на достаточное количество подсетей.

Поддержка немаршрутизируемых протоколов, таких как NetBIOS, NetBEUI или DEC LAT, которые не оперируют с таким понятием, как сеть. Маршрутизаторы могут обрабатывать пакеты таких протоколов двумя способами.

- В первом случае они могут работать с пакетами этих протоколов как мосты, то есть передавать их на основании изучения MAC - адресов. Маршрутизатор необходимо сконфигурировать особым способом, чтобы по отношению к некоторым немаршрутизируемым протоколам на некоторых портах он выполнял функции моста, а по отношению к маршрутизируемым протоколам - функции маршрутизатора. Такой мост/маршрутизатор иногда называют brouter (bridge плюс router).
- Другим способом передачи пакетов немаршрутизируемых протоколов является *инкапсуляция* этих пакетов в пакеты какого-либо сетевого протокола. Некоторые производители маршрутизаторов разработали собственные протоколы, специально предназначенные для инкапсуляции немаршрутизируемых пакетов. Кроме того, существуют стандарты для инкапсуляции некоторых протоколов в другие, в основном в IP. Примером такого стандарта является протокол DLSw, определяющий методы инкапсуляции пакетов SDLC и NetBIOS в IP-пакеты, а также протоколы PPTP и L2TP, инкапсулирующие кадры протокола PPP в IP-пакеты. Более подробно технология инкапсуляции рассматривается в главе, посвященной межсетевому взаимодействию.

Разделение функций построения и использования таблицы маршрутизации. Основная вычислительная работа проводится маршрутизатором при составлении таблицы маршрутизации с маршрутами ко всем известным ему сетям. Эта работа состоит в обмене пакетами протоколов маршрутизации, такими как RIP или OSPF, и вычислении оптимального пути к каждой целевой сети по некоторому критерию. Для вычисления оптимального пути на графе, как того требуют протоколы состояния связей, необходимы значительные вычислительные мощности. После того как таблица маршрутизации составлена, функция продвижения пакетов происходит весьма просто - осуществляется просмотр таблицы и поиск совпадения полученного адреса с адресом целевой сети. Если

совпадение есть, то пакет передается на соответствующий порт маршрутизатора. Некоторые маршрутизаторы поддерживают только функции продвижения пакетов по готовой таблице маршрутизации. Такие маршрутизаторы являются усеченными маршрутизаторами, так как для их полноценной работы требуется наличие полнофункционального маршрутизатора, у которого можно взять готовую таблицу маршрутизации. Этот маршрутизатор часто называется сервером маршрутов. Отказ от самостоятельного выполнения функций построения таблицы маршрутизации резко удешевляет маршрутизатор и повышает его производительность. Примерами такого подхода являются маршрутизаторы NetBuilder компании 3Com, поддерживающие фирменную технологию Boundary Routing, маршрутизирующие коммутаторы Catalyst 5000 компании Cisco Systems.

5.6.2. Корпоративные модульные концентраторы

Большинство крупных фирм-производителей сетевого оборудования предлагает модульные концентраторы в качестве «коммутационного центра» корпоративной сети. Такие концентраторы отражают тенденцию перехода от полностью распределенных локальных сетей 70-х годов на коаксиальном кабеле к централизованным коммуникационным решениям, активно воздействующим на передачу пакетов между сегментами и сетями. Модульные корпоративные концентраторы представляют собой многофункциональные устройства, которые могут включать несколько десятков модулей различного назначения: повторителей разных технологий, коммутаторов, удаленных мостов, маршрутизаторов и т. п., которые объединены в одном устройстве с модулями-агентами протокола SNMP, и, следовательно, позволяют централизованно объединять, управлять и обслуживать большое количество устройств и сегментов, что очень удобно в сетях большого размера.

Модульный концентратор масштаба предприятия обычно обладает внутренней шиной или набором шин очень высокой производительности - до нескольких десятков гигабит в секунду, что позволяет реализовать одновременные соединения между модулями с высокой скоростью, гораздо большей, чем скорость внешних интерфейсов модулей. Основная идея разработчиков таких устройств заключается в создании программно настраиваемой конфигурации связей в сети, причем сами связи между устройствами и сегментами могут также поддерживаться с помощью различных методов: побитовой передачи данных повторителями, передачи кадров коммутаторами и передачи пакетов сетевых протоколов маршрутизаторами.

Пример структуры корпоративного концентратора приведен на рис. 5.30. Он имеет несколько шин для образования независимых разделяемых сегментов Ethernet 10 Мбит/с, Token Ring и FDDI, а также высокоскоростную шину в 10 Гбит/с для передач кадров и пакетов между модулями коммутации и маршрутизации. Каждый из модулей имеет внешние интерфейсы для подключения конечных узлов и внешних коммуникационных устройств - повторителей, коммутаторов, а также несколько интерфейсов с внутренними шинами концентратора. Концентратор рассчитан на подключение конечных узлов в основном к внешним интерфейсам повторителей (для образования разделяемых сегментов) и коммутаторов (для поддержки микросегментации). Уже готовые сегменты, то есть образованные внешними повторителями и коммутаторами, могут подключаться к внешним интерфейсам коммутаторов и маршрутизаторов корпоративного концентратора. Дальнейшее соединение разделяемых сегментов и коммутируемых узлов и сегментов происходит модулями коммутации и маршрутизации концентратора по внутренним связям с помощью высокоскоростной шины. Конечно, модули могут связываться между

собой и через внешние интерфейсы, но такой способ нерационален, так как скорость обмена ограничивается при этом скоростью протокола интерфейса, например 10 Мбит/с или 100 Мбит/с. Внутренняя же шина соединяет модули на гораздо более высокой скорости, примерно 10/N Гбит/с, где N - количество портов, одновременно требующих обмена. Внешние связи между модулями превращают корпоративный концентратор просто в стойку с установленными модулями, а внутренний обмен делает эту стойку единым устройством с общей системой программного управления трафиком. Обычно для конфигурирования модулей и связей между ними производители корпоративных концентраторов сопровождают их удобным программным обеспечением с графическим интерфейсом. Отдельный модуль управления выполняет общие для всего концентратора функции: управления по протоколу SNMP, согласование таблиц коммутации и маршрутизации в разных модулях, возможно использование этого модуля как межмодульной коммутационной фабрики вместо общей шины.

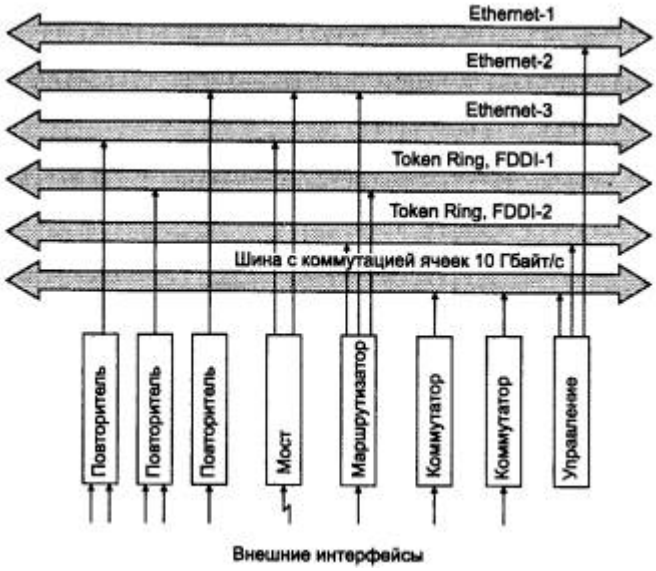


Рис. 5.30. Структура корпоративного модульного концентратора

Примерами корпоративных многофункциональных концентраторов могут служить устройства System 5000 компании Nortel Networks, MMAC-Plus компании Cabletron Systems, CoreBuilder 6012 компании 3Com.

Ввиду того, что отказ корпоративного модульного концентратора приводит к очень тяжелым последствиям, в их конструкцию вносится большое количество средств обеспечения отказоустойчивости.

5.6.3. Стирание граней между коммутаторами и маршрутизаторами

В классическом понимании терминов коммутатор - это устройство, принимающее решение о продвижении пакетов на основании заголовков протоколов 2-го уровня, то есть протоколов типа Ethernet или FDDI, а маршрутизатор - устройство, принимающее аналогичное решение на основании заголовков протоколов 3-го уровня, то есть уровня протоколов IP или IPX. В настоящее время наблюдается отчетливая тенденция по совмещению в одном устройстве функций коммутатора и маршрутизатора.

Соотношение коммутации и маршрутизации в корпоративных сетях

До недавнего времени сложившимся информационным потокам корпоративной сети наилучшим образом соответствовала следующая иерархическая структура. На нижнем уровне (уровне отделов) располагались сегменты сети, построенные на быстро работающих повторителях и коммутаторах. Сегменты включали в себя как рабочие станции так и серверы. В большинстве случаев было справедливо эмпирическое соотношение 80/20, в соответствии с которым основная часть трафика (80 %) циркулировала внутри сегмента, то есть порождалась запросами пользователей рабочих станций к серверам своего же сегмента.

На более высоком уровне располагался маршрутизатор, к которому подключалось сравнительно небольшое количество внутренних сетей, построенные на коммутаторах. Через порты маршрутизатора проходил трафик обращений рабочих станций одних сетей к серверам других сетей. Известно, что маршрутизатор затрачивает больше времени на обработку каждого пакета, чем коммутатор, поскольку он выполняет более сложную обработку трафика, включая интеллектуальные алгоритмы фильтрации, выбор маршрута при наличии нескольких возможных путей и т. п. С другой стороны, трафик, проходящий через порты маршрутизатора был менее интенсивный, чем внутрисегментный, поэтому сравнительно низкая производительность маршрутизатора не делала его узким местом.

Сегодня ситуация в корпоративных сетях быстро меняется. Количество пользователей стремительно растет. Пользователи избавляются от устаревающих текстовых приложений, отдавая предпочтение Web-интерфейсу. А завтра эти же пользователи будут работать с аудио, видео, push и другими, абсолютно новыми приложениями, основанными на новых технологиях распространения пакетов, таких как IP Multicast и RSVP. Не работает и старое правило 80/20, сегодня большое количество информации берется из публичных серверов Internet, а также из Web-серверов других подразделений предприятия, создавая большой межсетевой трафик. Существующие сети не оптимизировались для таких непредсказуемых потоков тра-фика, когда каждый может общаться почти с каждым. А с проникновением в корпоративные сети технологии Gigabit Ethernet эта проблема обострится еще больше.

Таким образом, сегодня образовался большой разрыв между производительностью типичного маршрутизатора и типичного коммутатора. В этой ситуации возможны два решения: либо отказаться вообще от маршрутизации, либо увеличить ее производительность.

Отказ от маршрутизации

За последние годы основные усилия были сосредоточены в первом направлении: применять маршрутизацию как можно реже, только там, где от нее никак нельзя отказаться. Например, на границе между локальной и глобальной сетью. Отказ от маршрутизаторов означает переход к так называемой плоской сети, то есть сети, построенной только на коммутаторах, а значит, и отказ от всех интеллектуальных возможностей обработки трафика, присущих маршрутизаторам. Такой подход повышает производительность, но приводит к потере всех преимуществ, которые давали маршрутизаторы, а именно:

- маршрутизаторы более надежно, чем коммутаторы, изолируют части большой составной сети друг от друга, защищая их от ошибочных кадров, порождаемых неисправным программным или аппаратным обеспечением других сетей (например, от широковещательных штормов);

- маршрутизаторы обладают более развитыми возможностями защиты от несанкционированного доступа за счет функций анализа и фильтрации трафика на более высоких уровнях: сетевом и транспортном;
- сеть, не разделенная маршрутизаторами, имеет ограничения на число узлов (для популярного протокола IP это ограничение составляет 255 узлов для сетей самого доступного класса С).

Из этого следует, что в сети необходимо сохранять функции маршрутизации в привычном смысле этого слова.

Что касается второго направления - повышение производительности маршрутизаторов, - сложилось так, что самые активные действия в этом направлении были предприняты производителями коммутаторов, наделявшими свои продукты некоторыми возможностями маршрутизаторов. Именно в модифицированных коммутаторах были впервые достигнуты скорости маршрутизации в 5-7 миллионов пакетов в секунду, а также опробованы многие важные концепции ускорения функций маршрутизации.

Коммутаторы 3-го уровня с классической маршрутизацией

Термин «коммутатор 3-го уровня» употребляется для обозначения целого спектра коммутаторов различного типа, в которые встроены функции маршрутизации пакетов. Функции коммутации и маршрутизации могут быть совмещены двумя способами.

- Классическим, когда маршрутизация выполняется по каждому пакету, требующему передачи из сети в сеть, а коммутация выполняется для пакетов, принадлежащих одной сети.
- Нестандартным способом ускоренной маршрутизации, когда маршрутизируется несколько первых пакетов устойчивого потока, а все остальные пакеты этого потока коммутируются.

Рассмотрим первый способ.

Классический коммутатор 3-го уровня подобно обычному коммутатору захватывает все кадры своими портами независимо от их MAC - адресов, а затем прини-. мает решение о коммутации или маршрутизации каждого кадра. Если кадр имеетMAC - адрес назначения, отличный от MAC - адреса порта маршрутизатора, то этот кадр коммутируется. Если устройство поддерживает технику VLAN, то перед передачей кадра проверяется принадлежность адресов назначения и источника одной виртуальной сети.

Если же кадр направлен непосредственно MAC - адресу какого-либо порта маршрутизатора, то он маршрутизируется стандартным образом. Коммутатор 3-го уровня может поддерживать динамические протоколы маршрутизации, такие как RIP или OSPF, а может полагаться на статическое задание маршрутов или на получение таблицы маршрутизации от другого маршрутизатора.

Такие комбинированные устройства появились сразу после разработки коммутаторов, поддерживающих виртуальные локальные сети (VLAN). Для Связи VLAN требовался маршрутизатор. Размещение маршрутизатора в одном корпусе с коммутатором позволяло получить некоторый выигрыш в производительности, например, за счет исключения одного этапа буферизации пакета, когда он передается из коммутатора в маршрутизатор. Хотя такие устройства с равным успехом можно называть маршрутизирующими

коммутаторами или коммутирующими маршрутизаторами, за ними закрепилось название коммутаторов 3-го уровня.

Примерами таких коммутаторов могут служить хорошо известные коммутаторы LANplex (теперь CoreBuilder) 6000 и 2500 компании 3Com. В этих устройствах совместно используются специализированные большие интегральные микросхемы (ASIC), RISC- и CISC-процессоры. Микросхемы ASIC обеспечивают коммутацию пакетов и их первичный анализ при маршрутизации, RISC-процессоры выполняют основную работу по маршрутизации, а CISC-процессоры реализуют функции управления. За счет такого распараллеливания процесса функционирования подсистем коммутации и маршрутизации достигается достаточно высокий уровень производительности. Так, система CoreBuilder 2500, имеющая один блок коммутации/маршрутизации, способна маршрутизировать 98 тысяч IP-пакетов в секунду (без их потери) на полной скорости каналов связи. Более мощная система CoreBuilder 6000 по данным компании 3Com в конфигурации с 88 портами Fast Ethernet маршрутизирует до 3 миллионов пакетов в секунду.

Более быстродействующей реализацией данного подхода являются устройства, в которых функции маршрутизации перенесены из универсального центрального процессора в специализированные заказные микросхемы портов. При этом ускорение процесса маршрутизации происходит не только за счет распараллеливания работы между несколькими процессорами, но и за счет использования специализированных процессоров вместо универсальных процессоров типа Motorola или Intel. Примеры этого подхода - коммутатор CoreBuilder 3500 компании 3Com, маршрутизирующий коммутатор Accelar 1200 компании Nortel Networks.

По данным фирм-производителей, коммутаторы 3-го уровня CoreBuilder 3500 и Accelar 1200 способны маршрутизировать соответственно до 4 и 7 миллионов пакетов в секунду. С такой же скоростью они коммутируют поступающие кадры, что говорит о высокой эффективности реализованных в ASIC алгоритмах маршрутизации.

Подход, связанный с переносом процедур маршрутизации из программируемых процессоров, пусть и специализированных, в работающие по жестким алгоритмам БИС, имеет один принципиальный недостаток - ему недостает гибкости. При необходимости изменения протокола или набора протоколов требуется перепроектировать БИС, что очевидно подразумевает очень большие затраты времени и средств по сравнению с изменением программного обеспечения маршрутизатора. Поэтому быстродействующие маршрутизаторы переносят в БИС только несколько базовых протоколов сетевого уровня, чаще всего IP и IPX, делая такие маршрутизаторы узко специализированными.

Маршрутизация потоков

Еще один тип коммутаторов 3-го уровня - это коммутаторы, которые ускоряют процесс маршрутизации за счет выявления устойчивых потоков в сети и обработки по схеме маршрутизации только нескольких первых пакетов потока. Многие фирмы разработали подобные схемы, однако до сих пор они являются нестандартными, хотя работа над стандартизацией этого подхода идет в рамках одной из рабочих групп IETF. Существуют компании, которые считают эти попытки ошибочными, вносящими ненужную путаницу в и так непростую картину работы стека протоколов в сети. Наиболее известной компанией, занявшей такую позицию, является компания Nortel Networks, маршрутизаторы которой Accelar 1200 работают по классической схеме. Тем не менее количество компаний, разработавших протоколы ускоренной маршрутизации, в основном ускоренной IP-

маршрутизации, довольно велико, туда входят такие известные компании, как 3Com, Cisco, Cabletron, Digital, Ipsilon.

Поток - это последовательность пакетов, имеющих некоторые общие свойства, по меньшей мере у них должны совпадать адрес отправителя и адрес получателя, и тогда их можно отправлять по одному и тому же маршруту. Желательно, чтобы пакеты потока имели одно и то же требование к качеству обслуживания.

Ввиду разнообразия предложенных схем опишем только основную идею, лежащую в их основе.

Если бы все коммутаторы/маршрутизаторы, изображенные на рис. 5.31, работали по классической схеме, то каждый пакет, отправляемый из рабочей станции, принадлежащей одной IP-сети, серверу, принадлежащему другой IP-сети, проходил бы через блоки маршрутизации всех трех устройств.

Рис. 5.31. Ускоренная маршрутизация потока пакетов

В схеме ускоренной маршрутизации такую обработку проходит только несколько первых пакетов долговременного потока, то есть классическая схема работает до тех пор, пока долговременный поток не будет выявлен.

После этого первый коммутатор на пути следования потока выполняет нестандартную обработку пакета - он помещает в кадр канального протокола, например Ethernet, не MAC - адрес порта следующего маршрутизатора, а MAC - адрес узла назначения, который на рисунке обозначен как $MAC_к$. Как только эта замена произведена, кадр с таким MAC - адресом перестает поступать на блоки маршрутизации второго и третьего коммутатора/маршрутизатора, а проходит только через блоки коммутации этих устройств. Процесс передачи пакетов действительно ускоряется, так как они не проходят многократно повторяющиеся этапы маршрутизации. В то же время защитные свойства маршрутизаторы сохраняют, так как первые пакеты проверяются на допустимость передачи в сеть назначения, поэтому сохраняются фильтрация широковещательного шторма, защита от несанкционированного доступа и другие преимущества сети, разделенной на подсети.

Для реализации описанной схемы нужно решить несколько проблем. Первая - на основании каких признаков определяется долговременный поток. Это достаточно легкая проблема, и основные подходы к ее решению очевидны - совпадение адресов и портов

соединения, общие признаки качества обслуживания, некоторый порог одинаковых пакетов для фиксации долговременное ™. Вторая проблема более серьезная. На основании какой информации первый маршрутизатор узнает MAC - адрес узла назначения. Этот узел находится за пределами непосредственно подключенных к первому маршрутизатору сетей, поэтому использование протокола ARP здесь не поможет. Именно здесь расходятся пути большинства фирменных технологий ускоренной маршрутизации. Многие компании разработали собственные служебные протоколы, с помощью которых маршрутизаторы запрашивают этот MAC - адрес друг у друга, пока последний на пути маршрутизатор не выяснит его с помощью протокола ARP.

Фирменные протоколы используют как распределенный подход, когда все маршрутизаторы равны в решении проблемы нахождения MAC - адреса, так и централизованный, когда в сети существует выделенный маршрутизатор, который помогает ее решить для всех.

Примерами коммутаторов 3-го уровня, работающими по схеме ускоренной IP-маршрутизации, являются коммутаторы Smart-Switch компании Cabletron, а также коммутатор Catalyst 5000 компании Cisco, выполняющий свои функции совместно с маршрутизаторами Cisco 7500 по технологии Cisco NetFlow для распознавания потоков и определения их адресной информации, и ряд других.

Выводы

- Типичный маршрутизатор представляет собой сложный специализированный компьютер, который работает под управлением специализированной операционной системы, оптимизированной для выполнения операций построения таблиц маршрутизации и продвижения пакетов на их основе.
- Маршрутизатор часто строится по мультипроцессорной схеме, причем используется симметричное мультипроцессирование, асимметричное мультипроцессирование и их сочетание. Наиболее рутинные операции обработки пакетов выполняются программно специализированными процессорами или аппаратно большими интегральными схемами (БИС/ASIC). Более высокоуровневые действия выполняют программно универсальные процессоры.
- По областям применения маршрутизаторы делятся на: магистральные маршрутизаторы, маршрутизаторы региональных подразделений, маршрутизаторы удаленных офисов и маршрутизаторы локальных сетей - коммутаторы 3-го уровня.
- Основными характеристиками маршрутизаторов являются: общая производительность в пакетах в секунду, набор поддерживаемых сетевых протоколов и протоколов маршрутизации, набор поддерживаемых сетевых интерфейсов глобальных и локальных сетей.
- · К числу дополнительных функций маршрутизатора относится одновременная поддержка сразу нескольких сетевых протоколов и нескольких протоколов маршрутизации, возможность приоритетной обработки трафика, разделение функций построения таблиц маршрутизации и продвижения пакетов между маршрутизаторами разного класса на основе готовых таблиц маршрутизации.
- Основной особенностью коммутаторов 3-го уровня является высокая скорость выполнения операций маршрутизации за счет их перенесения на аппаратный уровень - уровень БИС/ASIC.
- Многие фирмы разработали собственные протоколы ускоренной маршрутизации долговременных потоков в локальных сетях, которые маршрутизируют только

несколько первых пакетов потока, а остальные пакеты коммутируют на основе MAC - адресов.

- Корпоративные многофункциональные концентраторы представляют собой устройства, в которых на общей внутренней шине объединяются модули разного типа - повторители, мосты, коммутаторы и маршрутизаторы. Такое объединение дает возможность программного конфигурирования сети с определением состава подсетей и сегментов вне зависимости от из физического подключения к тому или иному порту устройства.

Вопросы и упражнения

1. В чем состоит отличие задач, решаемых протоколами сетевого уровня в локальных и глобальных сетях?
2. Сравните таблицу моста/коммутатора с таблицей маршрутизатора. Каким образом они формируются? Какую информацию содержат? От чего зависит их объем?
3. Таблица маршрутизации содержит записи о сетях назначения. Должна ли она содержать записи обо всех сетях составной сети или только о некоторых? Если только о некоторых, то о каких именно?
4. Может ли в таблице маршрутизации иметься несколько записей о маршрутизаторах по умолчанию?
5. На рис. 5.32 изображен компьютер с двумя сетевыми адаптерами, к которым подсоединены сегменты сети. Компьютер работает под управлением Windows NT. Может ли компьютер А обмениваться данными с компьютером В?
   A. Да, всегда.
   B. Нет, всегда.
   C. Все зависит от того, как сконфигурирована система Windows NT.

   Может ли повлиять на ответ тот факт, что в сегментах используются разные канальные протоколы, например Ethernet и Token Ring?

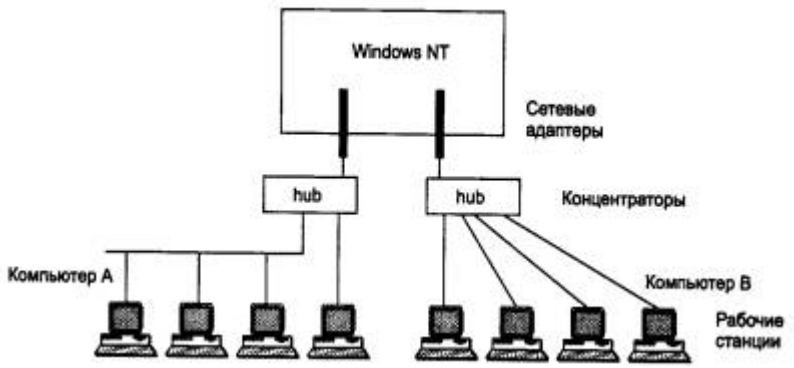


Рис. 5.32. Режимы работы компьютера с двумя сетевыми кортами

6. Сколько уровней имеет стек протоколов TCP/IP? Каковы их функции? Какие особенности этого стека обусловливают его лидирующее положение в мире сетевых технологий?
7. Какие протоколы стека TCP/IP относятся к уровню Internet (уровню межсетевого взаимодействия)?
8. В чем проявляется ненадежность протокола IP?

9. Могут ли быть обнаружены ошибки на уровне Internet? Могут ли они быть исправлены средствами этого уровня?
10. В чем особенности реализации алгоритма скользящего окна в протоколе TCP?
11. В составных сетях используются три вида адресов: символьные, сетевые и локальные. Какие из приведенных ниже адресов могли бы в составной IP-сети являться локальными, а какие нет?
    A. 6-байтовый MAC - адрес (например, 12-B3-3B-51-A2-10);
    B. адрес X.25 (например, 25012112654987);
    C. 12-байтовый IPX-адрес (например, 13.34.B4.0A.C5.10.11.32.54.C5.3B.01);
    D. адрес VPI/VCI сети ATM.
12. Какие из следующих утверждений верны всегда?
    A. Каждый порт моста/коммутатора имеет MAC - адрес.
    B. Каждый мост/коммутатор имеет сетевой адрес.
    C. Каждый порт моста/коммутатора имеет сетевой адрес.
    D. Каждый маршрутизатор имеет сетевой адрес.
    E. Каждый порт маршрутизатора имеет MAC - адрес.
    F. Каждый порт маршрутизатора имеет сетевой адрес.
13. Какую долю всего множества IP-адресов составляют адреса класса A? Класса B? Класса C?
14. Какие из ниже приведенных адресов не могут быть использованы в качестве IP-адреса конечного узла сети, подключенной к Internet? Для синтаксически правильных адресов определите их класс: A, B, C, D или E.
    A. 127.0.0.1
    B. 201.13.123.245
    C. 226.4.37.105
    D. 103.24.254.0
    E. 10.234.17.25
    F. 154.12.255.255
    G. 13.13.13.13
    H. 204.0.3.1
    I. 193.256.1.16
    J. 194.87.45.0
    K. 195.34.116.255
    L. 161.23.45.305
15. Пусть IP-адрес некоторого узла подсети равен 198.65.12.67, а значение маски для этой подсети - 255.255.255.240. Определите номер подсети. Какое максимальное число узлов может быть в этой подсети?
16. Пусть поставщик услуг Internet имеет в своем распоряжении адрес сети класса B. Для адресации узлов своей собственной сети он использует 254 адреса. Определите максимально возможное число абонентов этого поставщика услуг, если размеры требуемых для них сетей соответствуют классу C? Какая маска должна быть установлена на маршрутизаторе поставщика услуг, соединяющем его сеть с сетями абонентов?
17. Какое максимальное количество подсетей теоретически возможно организовать, если в вашем распоряжении имеется сеть класса C? Какое значение должна при этом иметь маска?
18. Почему даже в тех случаях, когда используются маски, в IP-пакете маска не передается?
19. Какие преимущества дает технология CIDR? Что мешает ее широкому внедрению?
20. Имеется ли связь между длиной префикса пула IP-адресов и числом адресов, входящих в этот пул?