Лекция 3. Сетевое оборудование.

3.1. Повторитель (концентратор, hub)

В начале 80-х годов сети Ethernet организовывались на базе шинной топологии с использованием сегментов на основе коаксиального кабеля. С увеличением длины кабеля, соединяющего компьютеры, усиливается затухание сигнала в кабеле, поэтому максимальная длина кабеля соединяющего компьютеры в сети не может превышать 500 метров для толстого жесткого коаксиального кабеля и 185 метров для тонкого кабеля Ethernet. Таким образом, максимально возможная общая длина всех кабелей сети – 500 метров. Для преодоления 500-метрового барьера используют повторители (repeater). Повторитель просто по битам копирует (пересылает) все пакеты Ethernet из одного сегмента сети во все другие, подключенные к нему (см. рис.1).

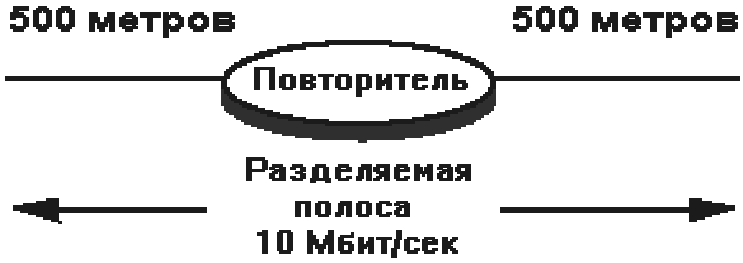


Рис. 1. Двухпортовый повторитель (схема).

Повторитель работает на физическом уровне модели OSI. Основной задачей повторителя является восстановление электрических сигналов для передачи их в другие сегменты. За счет усиления и восстановления формы электрических сигналов повторителем, становится возможным расширение сетей, построенных на основе коаксиального кабеля и увеличение общего числа пользователей сети.

Повторители бывают 2-х и многопортовыми. Двухпортовые повторители (см. рисунок 1) используются в сетях с шинной топологией, построенных на коаксиальном кабеле. Многопортовые повторители используются в сетях с топологией типа "звезда" (кабель "витая пара"). И 2-х и многопортовые повторители, получив пакет на один из своих портов, просто передает его во все остальные порты.

Многопортовые повторители, в сетях построенных на кабеле "витая пара", часто называют концентраторами или хабами (Hub). Хабы нужны даже не столько для усиления сигнала, как для соединения в сеть более чем двух компьютеров, т.к. кабель "витая пара" позволяет напрямую соединить только два компьютера. Если же необходимо соединить три компьютера, то каждый из них напрямую подключается к хабу, который и ретранслирует сигнал, полученный от одного компьютера на все остальные порты, к которым подключены другие компьютеры (см. рис. 2 и 3).

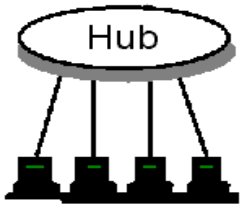


Рис. 2 Многопортовый повторитель (Hub) – схема подключения компьютеров по топологии "звезда".

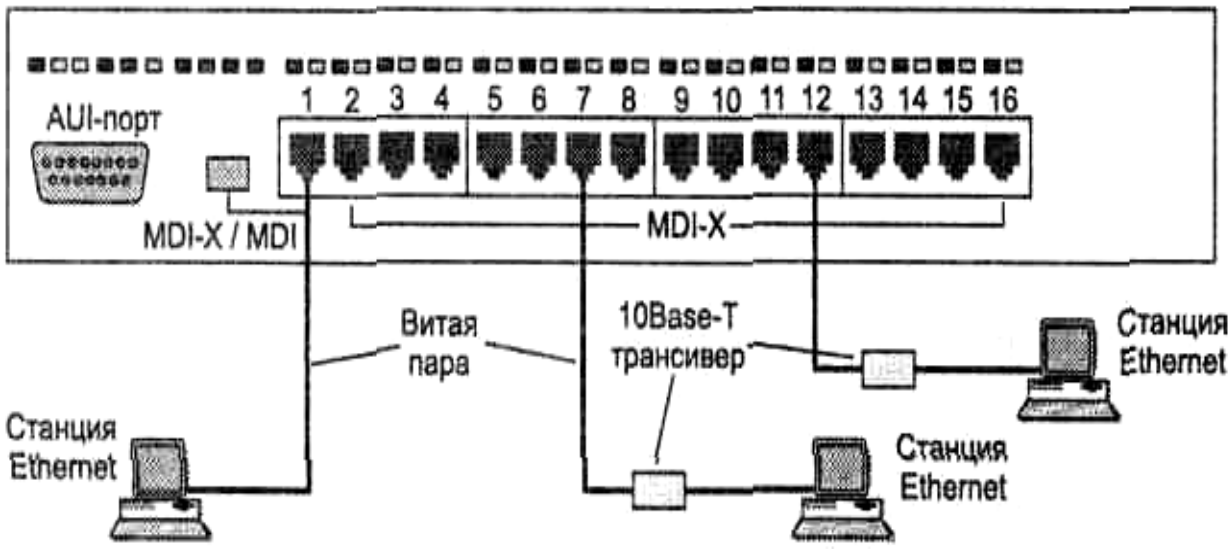


Рис. 3 Концентратор (hub) – внешний вид (схема).

Различия в моделях концентраторов при выполнении основной функции (побитное дублирование сигнала на все порты) невелики и, в основном, зависит от типа кабеля (витая пара, оптоволоконный и т.п.). Однако различные модели концентраторы могут реализовывать и дополнительные функции, некоторые из которых рассмотрены ниже.

Отключение портов

Концентраторы способны отключать некорректно работающие порты, изолируя тем самым остальную часть сети от возникших в узле проблем. Эту функцию называют *автосегментацией (autopartitioning).* Отключение происходит при отсутствии ответа на последовательность импульсов link test (проверка связи), посылаемых во все порты каждые 16 мс. В этом случае неисправный порт переводится в состояние "отключен", но импульсы link test будут продолжать посылаться в порт с тем, чтобы при восстановлении устройства работа с ним была продолжена автоматически. Отключение порта может произойти и по другим причинам:

- *Ошибки на уровне кадра.* Если интенсивность прохождения через порт кадров, имеющих ошибки, превышает заданный порог, то порт отключается, а затем, при отсутствии ошибок в течение заданного времени, включается снова. Такими ошибками могут быть: неверная контрольная сумм, неверная длина кадра (больше 1518 байт или меньше 64 байт), неоформленный заголовок кадра.
- *Множественные коллизии.* Если концентратор фиксирует, что источником коллизии был один и тот же порт 60 раз подряд, то порт отключается. Через некоторое время порт снова будет включен.
- *Затянувшаяся передача (jabber).* Если время прохождения одного кадра через порт превышает время передачи кадра максимальной длины в 3 раза, то порт отключается.

Поддержка резервных связей

Так как использование резервных связей в концентраторах определено только в стандарте FDDI, то для остальных стандартов разработчики концентраторов поддерживают эту функцию с помощью своих частных решений. Например, в концентраторах Ethernet/Fast Ethernet резервные связи всегда должны соединять отключенные порты, чтобы не нарушать логику работы сети. При конфигурировании концентратора администратор должен определить, какие порты являются основными, а какие по отношению к ним — резервными. Если по какой-либо причине порт отключается (срабатывает механизм автосегментации), концентратор делает активным его резервный порт. В некоторых моделях концентраторов разрешается использовать механизм назначения резервных портов только для оптоволоконных портов, считая, что нужно резервировать только наиболее важные связи, которые обычно выполняются на оптическом кабеле.

Защита от несанкционированного доступа

Общая разделяемая среда предоставляет очень удобную возможность для несанкционированного прослушивания сети и получения доступа к передаваемым данным. Для этого достаточно подключить компьютер с программным анализатором протоколов (сниффером - sniffer) к свободному разъему концентратора, записать на диск все проходящие по сети кадры, а затем выделить из них нужную информацию.

Разработчики концентраторов предоставляют различные способы защиты данных в разделяемых средах. Наиболее простой способ защиты заключается в том, что администратор вручную связывает с каждым портом концентратора некоторый MAC-адрес. Этот MAC-адрес является адресом сетевой карты компьютера, которому разрешено подключаться к данному порту. Например, на рис. первому порту концентратора назначен MAC-адрес 01:23 (условная запись). Компьютер с MAC-адресом сетевой карты 01:23 нормально работает с сетью через данный порт. Если злоумышленник отсоединяет этот компьютер и присоединяет вместо него свой, концентратор заметит, что при старте нового компьютера в сеть начали поступать кадры с адресом источника 07:89. Так как этот адрес является недопустимым для первого порта, то эти кадры фильтруются, порт отключается, а факт нарушения прав доступа может быть зафиксирован.

Другим способом защиты данных является случайное искажение данных в кадрах, передаваемых портам с адресом, отличным от адреса назначения пакета. При этом методе каждому порту концентратора также ставится в соответствие некоторый MAC-адрес сетевой карты. Кадр, поступивший на концентратор, дублируется на все порты, как этого и требует стандарт. При этом заголовки сдублированных кадров остаются неизменными, а в поле данных кадров помещаются нули. Полезные данные сохраняются только в поле данных кадра, направленного на порт, к которому подключен компьютер-адресат. Этот метод сохраняет логику случайного доступа к среде, так как все компьютеры видят, что сеть занята кадром, предназначенным одному из них (заголовок кадра не заполняется нулями), но только станция, которой послан этот кадр, может понять содержание поля данных кадра (см. рис.).

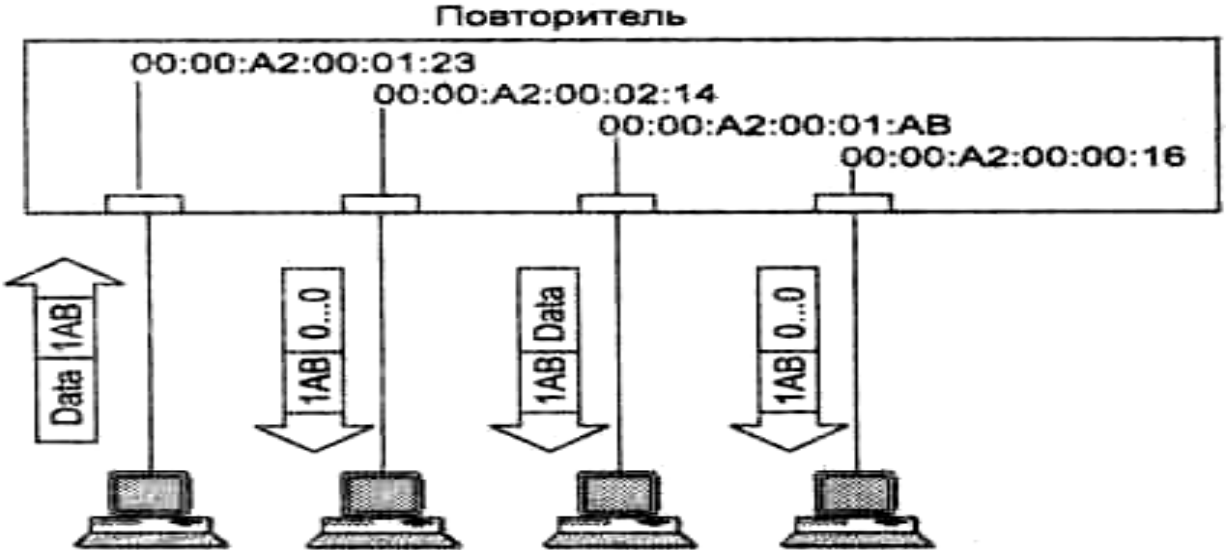


Рис. Искажение поля данных в кадрах, не предназначенных для приема станциями

Для реализации описанных выше методов защиты концентратор нужно предварительно сконфигурировать. Для этого концентратор должен иметь блок управления. Концентраторы, имеющие блок управления, обычно называют интеллектуальными (smart-hub). Блок управления представляет собой компактный вычислительный блок со встроенным программным обеспечением. Для взаимодействия администратора с блоком управления концентратор имеет консольный порт (чаще всего RS-232), к которому подключается терминал или персональный компьютер с программой эмуляции терминала. При присоединении терминала блок управления выдает на экран некоторое меню, с помощью которого администратор выбирает нужное действие и конфигурирует концентратор.

Многосегментные концентраторы

Многосегментные концентраторы обычно имеют большое количество портов (например, 72 или 240). Очевидно, что разделять среду передачи данных между таким количеством компьютеров нерационально. Поэтому в таких концентраторах имеется несколько несвязанных внутренних шин передачи данных, которые предназначены для создания нескольких разделяемых сред. Например, концентратор, изображенный на рис. , имеет три внутренние шины Ethernet. Первые два компьютера связаны с шиной Ethernet 3, а третий и четвертый компьютеры — с шиной Ethernet 1. Первые два компьютера образуют один разделяемый сегмент, а третий и четвертый — другой разделяемый сегмент.

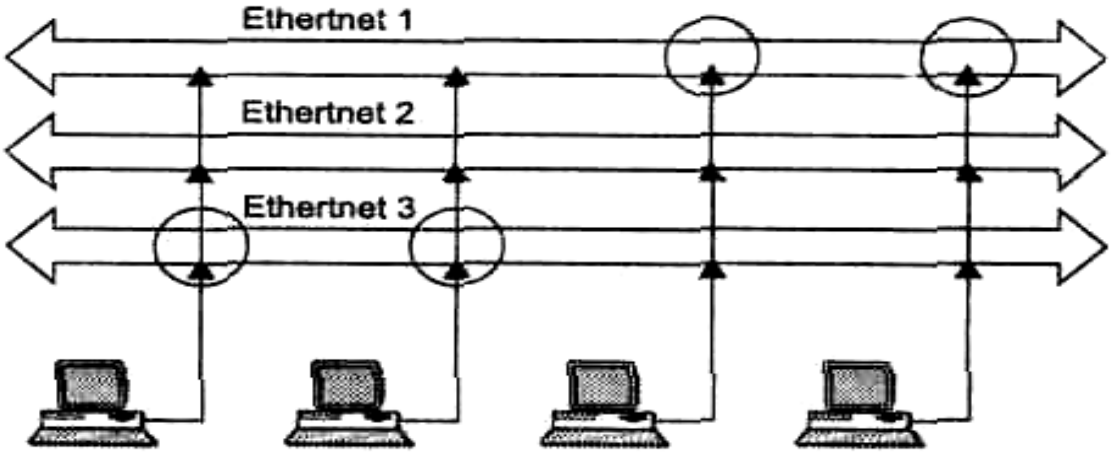


Рис. Многосегментный концентратор.

Между собой компьютеры, подключенные к разным сегментам, общаться через концентратор не могут, так как шины внутри концентратора никак не связаны. Для объединения сегментов необходимо использовать дополнительные сетевые устройства (мосты, коммутаторы, маршрутизаторы – см. дальше в лекциях). Многосегментные концентраторы нужны для создания разделяемых сегментов, состав которых может легко изменяться. Большинство многосегментных концентраторов, например System 5000 компании Nortel Networks или PortSwitch Hub компании 3Com, позволяют выполнять операцию соединения порта с одной из внутренних шин чисто программным способом, например с помощью локального конфигурирования через консольный порт. В результате администратор сети может присоединять компьютеры пользователей к любым портам концентратора, а затем с помощью программы конфигурирования концентратора управлять составом каждого сегмента. Если завтра сегмент Ethernet1 станет перегруженным, то его компьютеры можно распределить между оставшимися сегментами концентратора. Возможность многосегментного концентратора программно изменять связи портов с внутренними шинами называется конфигурационной коммутацией (configuration switching).

Управление концентратором по протоколу SNMP

Как видно из описания дополнительных функций, многие из них требуют конфигурирования концентратора. Это конфигурирование может производиться локально, путем подключения персонального компьютера или терминала к концентратору через интерфейс RS-232C, однако при большом количестве концентраторов в сети это становится неудобным. Поэтому большинство концентраторов, поддерживающих интеллектуальные дополнительные функции, могут управляться централизованно по сети с помощью протокола управления сетью SNMP (Simple Network Management Protocol) из стека TCP/IP.

В блок управления концентратором встраивается так называемый SNMP-агент, который имеет свой MAC- и IP-адрес. SNMP-агент собирает информацию о состоянии концентратора и хранит ее в базе данных управляющей информации — Management Information Base (MIB) – блока управления, которая позволяет одному из компьютеров сети, выполняющему роль центральной станции управления, запрашивать у SNMP-агента значения стандартных переменных базы MIB. В переменных хранятся не только данные о состоянии концентратора, но и управляющая информация, воздействующая на него. Например, в MIB есть переменная, управляющая состоянием порта ("включить" – "выключить").

Конструктивное исполнение концентраторов

По конструктивным особенностям выделяют следующие типы концентраторов:

- концентраторы с фиксированным количеством портов
- модульные концентраторы
- стековые концентраторы
- модульно-стековые концентраторы

Концентратор с фиксированным количеством портов — это наиболее простое конструктивное исполнение, когда устройство представляет собой отдельный корпус со всеми необходимыми элементами (портами, органами индикации и управления, блоком питания), и эти элементы заменять нельзя. Обычно общее количество портов изменяется от 4-8 до 24. Один порт может быть специально выделен для подключения концентратора к другому концентратору или иметь кнопочный переключатель, позволяющий подключать к этому порту как обычный компьютер (маркировка MDI-X, см. рис. 3), так и другой концентратор (маркировка MDI). Концентратор также может иметь разъем AUI для соединения (при помощи трансивера) с толстым коаксиальным кабелем, концентратором оптоволоконных сетей или другим концентратором "витая пара".

Модульный концентратор выполняется в виде отдельных модулей с фиксированным количеством портов, устанавливаемых на общее шасси. Шасси имеет внутреннюю шину для объединения отдельных модулей в единый повторитель. Часто такие концентраторы являются многосегментными, тогда в пределах одного модульного концентратора работает несколько несвязанных между собой повторителей. Агент протокола SNMP обычно выполняется в виде отдельного модуля, при установке которого концентратор превращается в интеллектуальное устройство. Модульные снабжаются системой терморегулирования, избыточными источниками питания, позволяют осуществлять замену модулей без отключения питания и дают возможность быстро и гибко реагировать на изменения конфигурации сети. Недостатком модульных концентраторов на основе шасси является высокая начальная стоимость такого устройства, т.к. даже если установлено всего 1-2 модуля, концентратор поставляется вместе со всеми общими устройствами (избыточные источники питания и т. п). Поэтому для сетей средних размеров большую популярность завоевали стековые концентраторы.

Стековый концентратор, как и концентратор с фиксированным числом портов, выполнен в виде отдельного корпуса с фиксированным количеством портов. Однако стековыми эти концентраторы называются не потому, что они устанавливаются один на другой, в общую стойку. Стековые концентраторы имеют специальные порты и кабели для объединения нескольких корпусов в единый повторитель, который имеет общий блок повторения и, с точки зрения правила 4-х хабов, считается одним повторителем. Число объединяемых в стек корпусов может быть достаточно большим (обычно до 8, но бывает и больше). Выгодной чертой стековых концентраторов является их более низкая стоимость, так как сначала предприятие может купить

одно устройство без избыточного шасси, а потом нарастить стек еще несколькими аналогичными устройствами.

Модулъно-стековые концентраторы представляют собой модульные концентраторы, объединенные специальными кабелями в стек. Как правило, корпуса таких концентраторов рассчитаны на небольшое количество модулей (1-3). Эти концентраторы сочетают достоинства концентраторов обоих типов.

3.2. Мост (bridge)

Повторители, за счет усиления и восстановления формы электрических сигналов, позволяют увеличить протяженность сети, однако и здесь есть ограничения: из-за задержки приема-передачи сигнала в повторителе, между любыми двумя компьютерами в сети Ethernet не может быть более 4-х повторителей, а в сети Fast Ethernet – не более одного повторителя 1-го класса и не более двух повторителей 2-го класса (подробнее см. далее в лекциях). Поэтому для создания более протяженных сетей необходимо пользоваться дополнительными сетевыми устройствами – мостами (bridge).

Мосты позволяют преодолеть ограничение "не более четырех повторителей между любыми двумя компьютерами" за счет того, что работают не на физическом, а на канальном уровне модели OSI. Т.е. мост ретранслирует кадр не по битам, а полностью принимает кадр в свой буфер, ***заново*** получает доступ к разделяемой среде и ретранслирует кадр в сеть. Помимо увеличения протяженности сети, мост также позволяет разбить ее на сегменты с независимыми разделяемыми средами, увеличив общую пропускную способность сети. Поясним на примере: пусть имеется три повторителя (хаба), к каждому из которых, при помощи кабеля "витая пара", подключено по четыре компьютера (см. рис.). Повторители соединены между собой при помощи моста. Допустим компьютер К1 передает в сеть кадр сообщения для компьютера К4. Кадр сообщения по кабелю попадет на повторитель1, который сдублирует его на все свои порты, т.е. кадр сообщения получат компьютеры К2, К3, К4 (что и требовалось) и мост. Мост, получив кадр сообщения от повторителя, анализирует "адрес получателя", имеющийся в кадре, определяет что компьютер К4 относится к сегменту 1 и поэтому кадр сообщения не надо дублировать для повторителей 2 и 3 (если бы кадр сообщения относился к компьютеру К5, то мост передал бы этот кадр только повторителю 2, ничего не передавая на порт, к которому подключен повторитель 3).

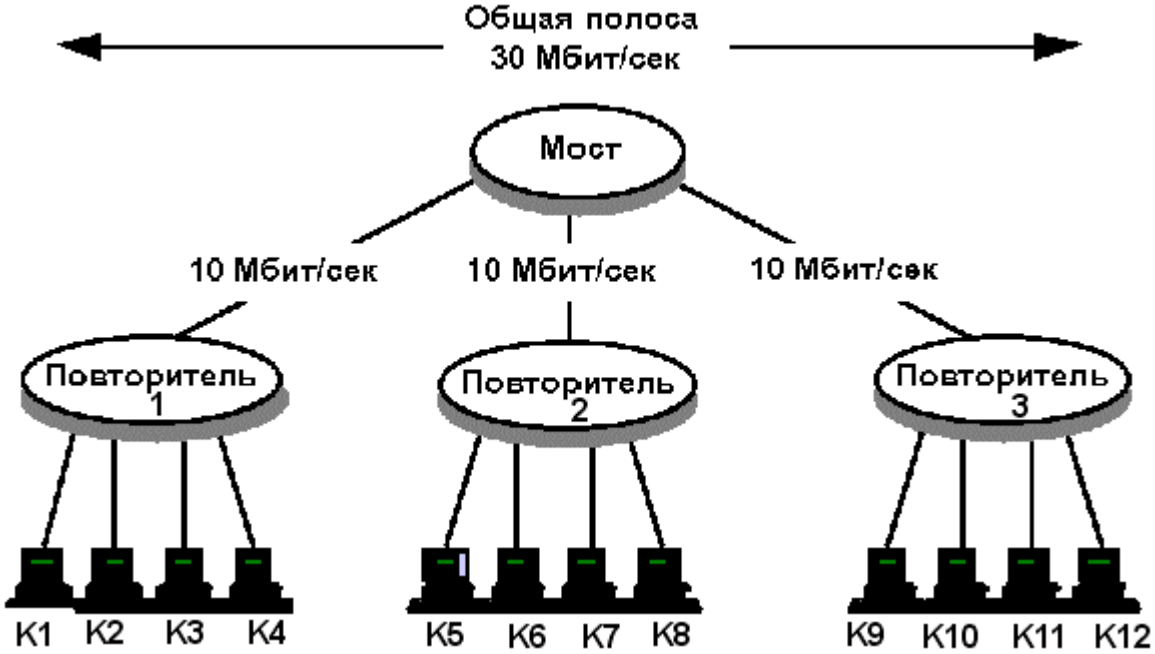


рис. Пояснение алгоритма работы моста.

Рассмотрим какие выгоды дает такая схема. В момент когда компьютер К1 передает кадр сообщения, ни один из компьютеров К2-К4 не может ничего передавать в сеть – сеть "занята". Однако в тот же момент времени компьютеры К5-К12 могут передавать сообщения друг другу – для них сеть "свободна", т.к. мост не передал кадр сообщения от компьютера К1 в их сегменты сети. Таким образом, если файл копируется с компьютера К1 на К4 со скоростью 10 Мбит/с, с компьютера К5 на К8 со скоростью 10 Мбит/с, с компьютера К9 на К11 со скоростью 10 Мбит/с, то суммарная пропускная способность сети составляет 30 Мбит/с. Если бы вместо моста в вершине этой схемы стоял простой повторитель, то кадр сообщения от компьютера К1 "занял" бы всю сеть, и ни один из компьютеров К2-К12 не смог бы в это время передавать в сеть что-либо (без возникновения коллизии), а пропускная способность сети упала бы до 10 Мбит/с.

Существует два основных алгоритма работы моста: алгоритм прозрачного моста и алгоритм моста с маршрутизацией от источника. Алгоритм прозрачного моста используется в сетях Ethernet, а алгоритм моста

с маршрутизацией от источника может использоваться в сетях Token Ring и FDDI, хотя в этих сетях могут использоваться и обычные прозрачные мосты.

Алгоритм работы прозрачного моста.
Мост при таком алгоритме не заметен (прозрачен) для сетевых карт. Сетевая карта посылает кадр данных сетевой карте другого компьютера так, как если бы моста в сети и не было. Порты моста подключены к соединяемым сегментам сети и не имеют своих MAC-адресов, захватывая все проходящие по сети пакеты. Первоначально мост не знает к какому порту подключены какие компьютеры (см. рис.). Поэтому, если компьютер 1 направит кадр компьютеру 2, то мост, захватив этот пакет на порту 1, сдублирует его на все остальные порты, т.е. в данном случае на порт 2 (хотя по логике работы моста этого делать и не надо, но мост пока не знает к какому сегменту относится компьютер 2). Одновременно с этим, мост сделает в своей внутренней таблице адресов запись, что компьютер 1 относится к сегменту 1 (т.к. кадр от него был захвачен с порта 1), и кадры для компьютера 1 надо дублировать только на порт 1. Если все четыре компьютера достаточно активны в сети, то таблица адресов моста заполнится, и он будет дублировать кадры только на те порты, на которые это действительно необходимо. Таким образом, трафик между компьютерами 1 и 2 будет отделен от трафика между компьютерами 3 и 4, т.е. кадры от компьютера 1 к компьютеру 2 не будут дублироваться на порт 2.

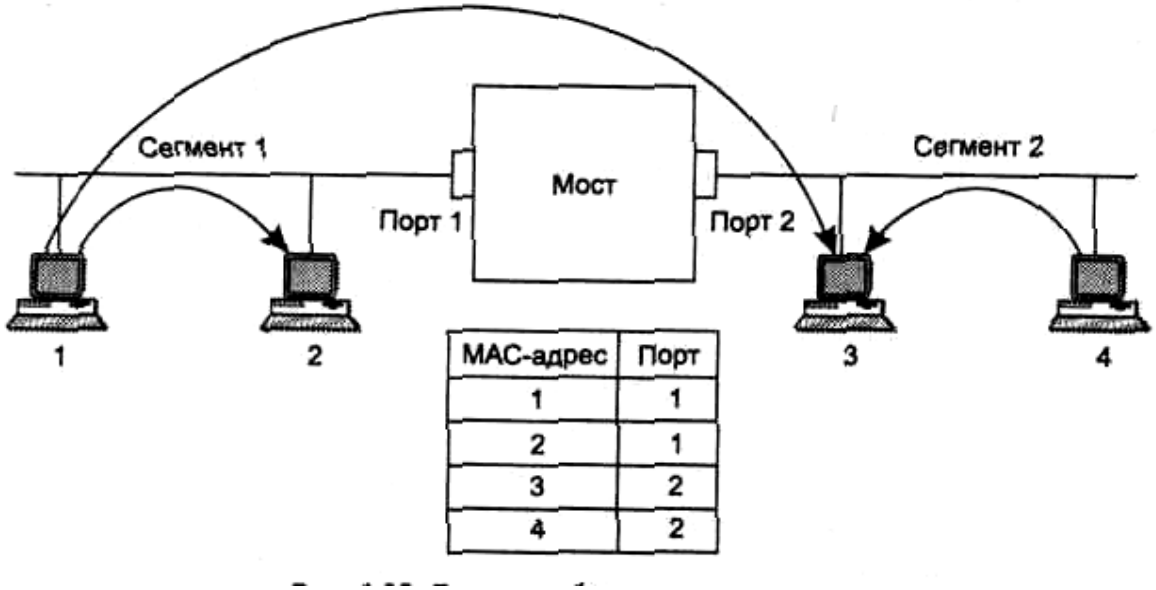


рис. Алгоритм работы прозрачного моста

Каждая автоматически созданная запись о принадлежности компьютера к сегменту 1 или 2 имеет срок жизни. Если до истечения срока жизни запись не подтверждалась кадрами, проходящими по сети, то она помечается как недействительная. Если, в любой момент времени, компьютер 1 будет перемещен в сегмент 2 и пакеты от него станут поступать на порт 2, то соответствующая запись в таблице будет изменена. Помимо динамически создаваемых записей, могут существовать и статические записи, созданные администратором вручную, при конфигурировании моста, и не имеющие срока жизни. При помощи статических записей можно жестко описать принадлежность компьютера к тому или иному сегменту, или указать, что пакеты к компьютеру 1 должны всегда дублироваться на все порты (flood - затопление), а пакеты к компьютеру 2 никогда не должны дублироваться ни на какие порты (discard - отбросить).

Алгоритм работы моста с маршрутизацией от источника (SR-мосты).
Этот алгоритм используется в сетях Token Ring и FDDI. Компьютер-отправитель помещает в кадр всю адресную информацию о промежуточных мостах и кольцах, которые кадр должен пройти на пути к компьютеру-адресату. Первоначально компьютер-отправитель не имеет никакой информации о пути к компьютеру адресату. Кадр просто передается в кольцо, в надежде, что адресат находится в одном кольце с отправителем. Если компьютер-адресат в кольце отсутствует не так, то кадр сделает оборот по кольцу и вернется без установленного признака "кадр получен" (бит "адрес распознан" и бит "кадр скопирован"). В таком случае компьютер-отправитель пошлет одномаршрутный широковещательный кадр-исследователь (SRBF, Single Route Brodcast Frame). Этот кадр распространяется по сети: мосты дублируют кадр на все свои порты, за исключением заблокированных администратором (для избежания петлевых маршрутов и зацикливания кадра). В конце-концов кадр-иследователь будет получен компьютером-адресатом, который немедленно отправит многомаршрутный широковещательный кадр-исследователь (ARBF, All Route Brodcast Frame). Этот кадр распространяется по сети, дублируясь мостами на все порты без исключения (для предотвращения зацикливания, кадр-ARBF уже однажды сдублированный мостом на один из своих портов, заново

на этот порт не дублируется). В конце-концов, до компьютера-отправителя дойдет множество кадров-ARBF, прошедших через все возможные маршруты от компьютера-адресата до компьютера-исследователя. Полученная информация попадет компьютеру-отправителю и в маршрутные таблицы моста, соединяющего кольцо компьютера-отправителя с остальной сетью. Впоследствии все компьютеры этого кольца могут воспользоваться информацией моста при отправке своих кадров.

Ограничения топологии сетей, построенных на прозрачных мостах.

Основным ограничением при использовании мостов является отсутствие петлевых маршрутов. Поясним на примере. Пусть имеется сеть, изображенная на рис.

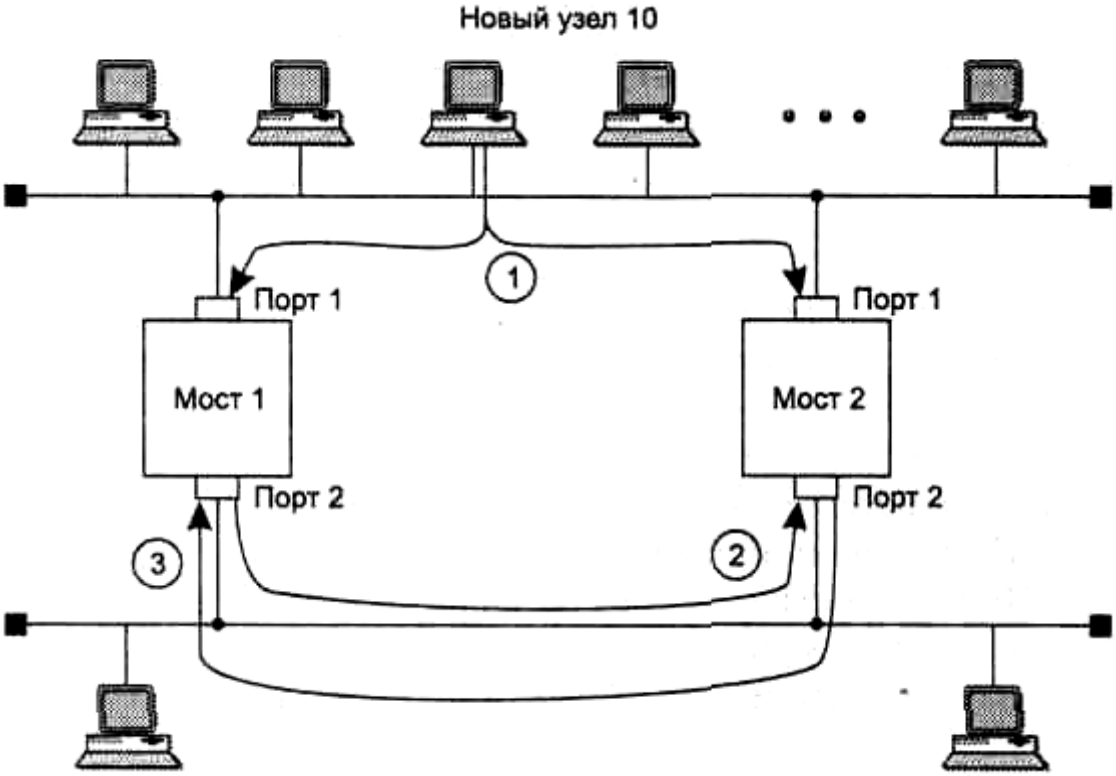


рис. Ошибки в работе мостов, возникающие при наличии петлеобразных маршрутов.

Пусть новый компьютер с адресом 10 впервые начинает работу в данной сети. Обычно начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых компьютер заявляет о своем существовании и одновременно ищет серверы сети. На этапе 1 компьютер посылает первый кадр с широковещательным адресом назначения и адресом источника 10 в свой сегмент. Кадр попадает как в мост 1, так и в мост 2. В обоих мостах новый адрес источника 10 заносится в адресную таблицу с пометкой о его принадлежности сегменту 1, то есть создается новая запись вида: MAC-адрес 10 – порт 1. Так как кадр, рассылаемый компьютером, имеет широковещательный адрес назначения, то каждый мост должен передать кадр на сегмент 2. Эта передача происходит поочередно, в соответствии с методом случайного доступа CSMA/CD технологии Ethernet. Пусть первым доступ к сегменту 2 получил мост 1 (этап 2). При ретрансляции мостом 1 кадра в сегмент 2 мост 2 принимает его в свой буфер и обрабатывает. Он видит, что адрес 10 уже есть в его адресной таблице, но пришедший кадр является более свежим, и он утверждает, что адрес 10 принадлежит сегменту 2, а не 1. Поэтому мост 2 корректирует содержимое базы и делает запись о том, что адрес 10 принадлежит сегменту 2. Аналогично поступает мост 1, когда мост 2 получит доступ к разделяемой среде и передает свою копию широковещательного кадра на сегмент 2. Результатом описанной ситуации является следующее:

- Размножение кадра, то есть появление нескольких его копий (в данном случае — двух, но если бы сегменты были соединены тремя мостами — то трех и т. д.).
- Бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком.
- Постоянная перестройка мостами своих адресных таблиц, так как кадр с адресом источника 10 будет появляться то на одном порту, то на другом.

Чтобы исключить все эти нежелательные эффекты, мосты нужно применять так, чтобы между логическими сегментами не было петель, то есть строить с помощью мостов только древовидные структуры, гарантирующие наличие только одного пути между любыми двумя сегментами. В простых сетях сравнительно легко гарантировать существование одного и только одного пути между двумя сегментами. Но когда количество

соединений возрастает и сеть становится сложной, то вероятность непреднамеренного образования петли оказывается высокой. Кроме того, желательно для повышения надежности иметь между мостами резервные связи, которые не участвуют при нормальной работе основных связей в передаче информационных пакетов станций, но при отказе какой-либо основной связи образуют новую связную рабочую конфигурацию без петель. Поэтому в сложных сетях между логическими сегментами прокладывают избыточные связи, которые образуют петли, но для исключения активных петель блокируют некоторые порты мостов. Наиболее просто эта задача решается вручную, но существуют и алгоритмы, которые позволяют решать ее автоматически. Наиболее известным является стандартный алгоритм покрывающего дерева (Spanning Tree Algorithm, STA). Кроме того, имеются фирменные алгоритмы, решающие ту же задачу, но с некоторыми улучшениями для конкретных моделей мостов и коммутаторов.

Удаленные мосты

Удаленный мост – это мост, который через один или несколько портов подключен к глобальной сети (Internet, X.25, FrameRelay, ATM). Удаленные мосты (а также удаленные маршрутизаторы) используются для соединения локальных сетей через глобальные сети. Если в локальных сетях мосты постепенно вытесняются коммутаторами, то удаленные мосты и сегодня продолжают пользоваться популярностью. Удаленные мосты не надо конфигурировать (адресная таблица строится автоматически), а при объединении с сетью предприятия сетей филиалов, где нет квалифицированного обслуживающего персонала, это свойство оказывается очень полезным.

Как и в локальных сетях, важной характеристикой удаленных мостов (удаленных маршрутизаторов) является скорость обработки кадров, которые часто ограничиваются не внутренними возможностями устройства, а скоростью передачи данных по линии (например, аналоговой телефонной линии). Для преодоления ограничений на скорость линии, а также для уменьшения части локального трафика, передаваемого по глобальной линии, в удаленных мостах и маршрутизаторах используются специальные приемы, отсутствующие в локальных устройствах. Эти приемы не входят в стандарты протоколов, но они реализованы практически во всех устройствах, обслуживающих низкоскоростные каналы, особенно каналы со скоростями в диапазоне от 9600 бит/с до 64 Кбит/с. К таким приемам относятся технологии сжатия пакетов, спуфинга и сегментации пакетов.

Сжатие пакетов (компрессия). Некоторые производители, используя собственные алгоритмы, обеспечивают коэффициент сжатия до 8:1. Стандартные алгоритмы сжатия, применяемые в модемах, обеспечивают коэффициент сжатия до 4:1. После сжатия данных для передачи требуется существенно меньшая скорость канала.

Спуфинг (spoofing). Эта технология позволяет значительно повысить пропускную способность линий, объединяющих локальные сети, работающие по протоколам с большим количеством широковещательных пакетов. Широковещательные пакеты характерны для большинства сетевых операционных систем, за исключением ОС Unix, которая изначально строилась для медленных глобальных линий связи. Главной идеей спуфинга является имитация передачи пакета по глобальной сети. Рассмотрим технику спуфинга на примере передачи между удаленными сетями пакетов SAP (Service Advertising Protocol) сервера ОС NetWare. Эти пакеты каждый сервер генерирует каждую минуту, чтобы все клиенты сети могли составить правильное представление об имеющихся в сети разделяемых ресурсах — файловых службах, службах печати и т. п. SAP-пакеты распространяются в IPX-пакетах с широковещательным сетевым адресом. Удаленные мосты должны передавать широковещательные пакеты на все свои порты (маршрутизаторы не должны передавать широковещательные пакеты из сети в сеть, но для SAP-пакетов сделано исключение — маршрутизатор, поддерживающий IPX, распространяет его на все порты, кроме того, на который этот пакет поступил). Таким образом, по выделенной линии может проходить достаточно большое количество SAP-пакетов. Если эти пакеты посылаются каким-либо сервером, но не доходят до клиентов, то клиенты не могут воспользоваться службами этого сервера. Если маршрутизаторы или мосты, объединяющие сети, поддерживают технику спуфинга, то они передают по выделенному каналу не каждый SAP-пакет, а например, только каждый пятый. Интенсивность служебного трафика в глобальном канале при этом уменьшается. Но для того, чтобы клиенты не теряли из списка ресурсов удаленной сети серверы, мост (маршрутизатор) имитирует приход этих пакетов по глобальному каналу, посылая SAP-пакеты от своего имени каждую минуту, как это и положено по протоколу. При этом мост (маршрутизатор) посылает несколько раз копию реального SAP-пакета, получаемого раз в 5 минут по выделенному каналу.

Сегментация пакетов — позволяет разделять большие передаваемые пакеты и передавать их сразу через две телефонные линии. Хотя это и не делает телефонные каналы более эффективными, но все же увеличивает скорость обмена данными почти вдвое.

3.3. Коммутатор (switch)

В последнее время наблюдается вытеснение мостов коммутаторы. Коммутаторы, как и мосты работают на канальном уровне и позволяют разделить общую разделяемую среду на несколько независимых сегментов передачи данных. Алгоритм работы коммутаторов аналогичен алгоритму работы прозрачного моста. Основным отличием, обеспечившим вытеснение мостов коммутаторами – это гораздо более высокая скорость работы коммутаторов. Мост должен полностью получить кадр данных перед тем, как ретранслировать его на соответствующий порт. Коммутатор начинает ретрансляцию кадра, не дожидаясь его полного получения (достаточно получить несколько первых байт кадра, содержащих адрес назначения). Кроме того, коммутатор позволяет организовать сразу несколько параллельных соединений между различными парами портов, что повышает пропускную способность сети в несколько раз. Однако коммутатор не может организовать одновременное соединение несколько портов – к одному порту (см. рис.).

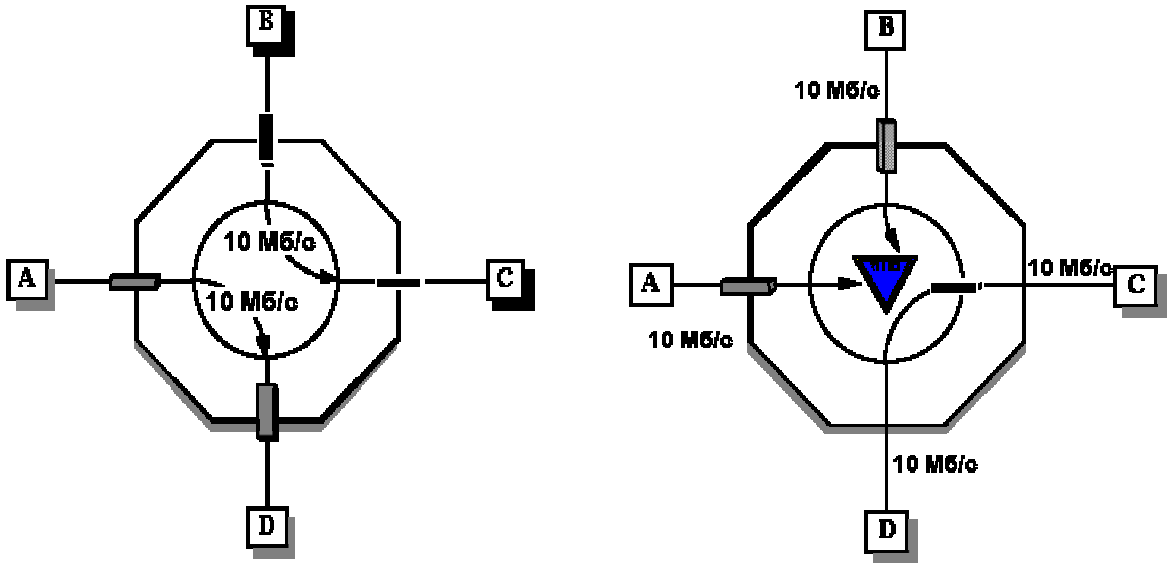
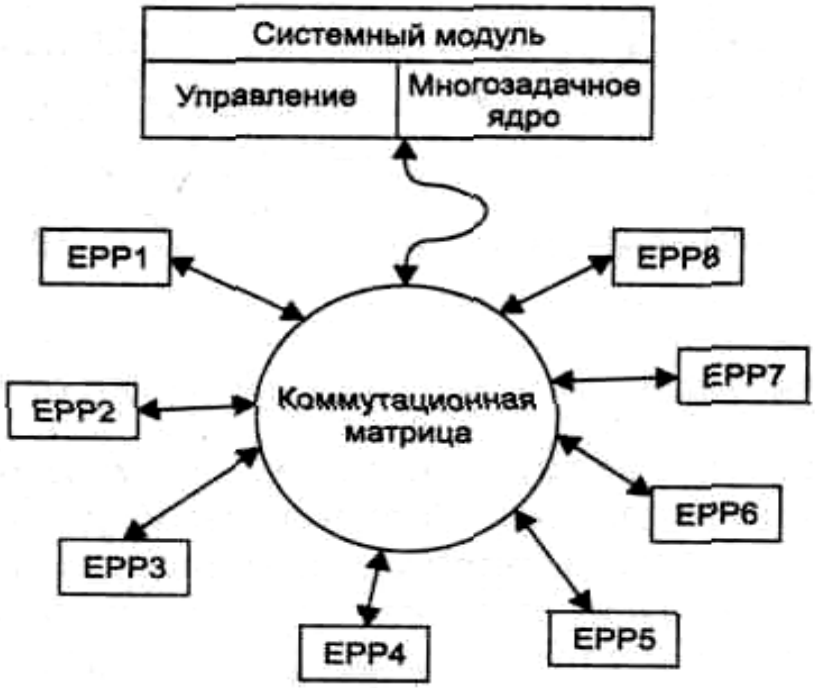


Рис. Параллельные соединения между портами коммутатора

Технология коммутаторов Ethernet была предложена фирмой Kalpana в 1990 году в ответ на растущие потребности в повышении пропускной способности сетей. Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana, представлена ниже (см. рис.).



Каждый из 8 портов коммутатора обслуживается собственным процессором пакетов Ethernet — EPP (Ethernet Packet Processor). Кроме того, коммутатор имеет системный модуль, который координирует работу всех процессоров EPP. Системный модуль ведет общую адресную таблицу коммутатора (какие компьютеры подключены к каким портам) и обеспечивает управление коммутатором по протоколу SNMP. Для передачи кадров между портами используется коммутационная матрица, подобная тем, которые работают в телефон-

ных коммутаторах или мультипроцессорных компьютерах. При поступлении кадра в какой-либо порт, процессор EPP буферизует несколько первых байт кадра, чтобы прочитать адрес назначения. После получения адреса назначения процессор сразу же принимает решение о передаче пакета, не дожидаясь прихода остальных байт кадра. Для этого он просматривает свой собственный кэш адресной таблицы, а если не находит там нужного адреса, обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров EPP. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку (адрес компьютера – номер порта), которая запоминается процессором EPP в своем кэше для последующего использования. После определения того, к какому порту подключен сегмент компьютера – адресата, процессор EPP обращается к коммутационной матрице и пытается установить соединение с нужным портом. Если порт занят, то кадр полностью буферизуется процессором EPP входного порта, после чего процессор ожидает освобождения выходного порта. После освобождения, данные передаются на выходной порт, который получает доступ к своему сегменту сети по методу CSMA/CD и передает кадр данных в свой сегмент.

Типы коммутаторов

По конструктивному исполнению выделяют следующие типы коммутаторов:

- коммутаторы с фиксированным количеством портов
- модульные коммутаторы на основе шасси
- стековые коммутаторы
- модульно-стековые коммутаторы

Различия между этими типами коммутаторов аналогичны различиям между соответствующими типами концентраторов (см. выше).

По способу коммутации портов в коммутаторе выделяют следующие типы коммутаторов:

- коммутаторы на основе коммутационной матрицы
- коммутаторы с общей шиной
- коммутаторы с разделяемой памятью
- комбинированные коммутаторы

Коммутаторы на основе коммутационной матрицы обеспечивают основной и самый быстрый способ взаимодействия процессоров портов. Однако реализация матрицы возможна только для определенного числа портов, причем сложность схемы возрастает пропорционально квадрату количества портов коммутатора. Чисто условно коммутационную матрицу можно представить следующим рисунком:

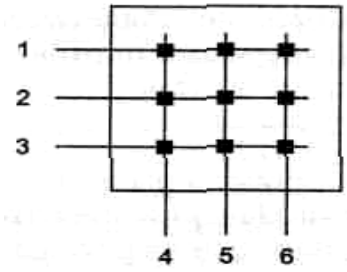


рис. Условная схема коммутационной матрицы.

Рассмотрим один из вариантов физической реализации коммутационной матрицы для 8 портов (см. рис.). Входные блоки процессоров EPP добавляют к байтам исходного кадра информацию о том на какой из портов его необходимо передать в виде специального ярлыка — тэга (tag). Для данного примера тэг представляет собой число их 3-х бит, соответствующее номеру выходного порта. Матрица состоит из трех уровней двоичных переключателей, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тэга. Переключатели первого уровня управляются первым битом тэга, второго — вторым, а третьего — третьим.

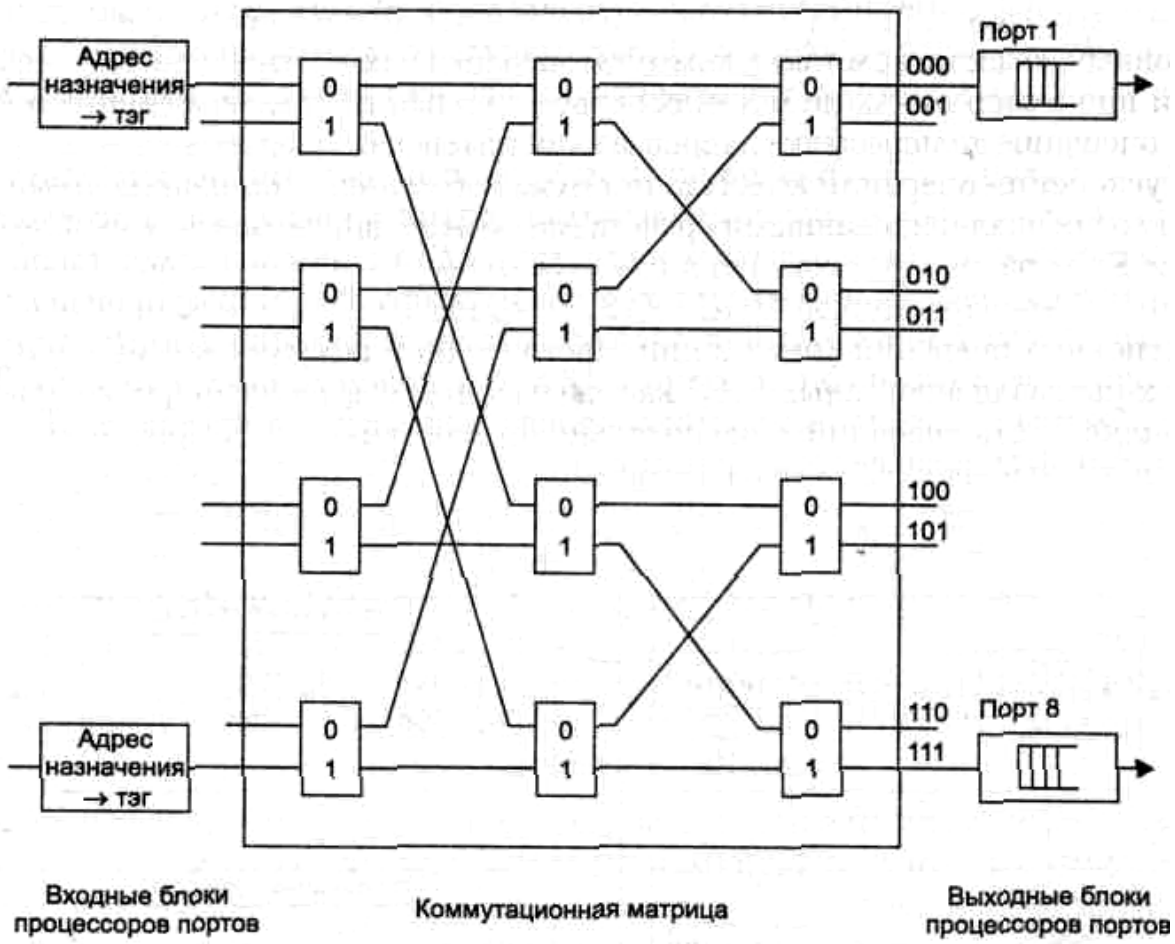


Рис. Реализация коммутационной матрицы 8x8 с помощью двоичных переключателей.

Основные достоинства таких матриц — высокая скорость коммутации портов и регулярная структура, которую удобно реализовывать в интегральных микросхемах. Недостатком является сложность наращивания числа портов и отсутствие буферизации данных внутри коммутационной матрицы (если порт занят, то данные должны накапливаться во входном блоке порта, принявшего кадр).

В коммутаторах с общей шиной процессоры портов связаны высокоскоростной шиной передачи данных, используемой в режиме разделения времени (см. рис.).

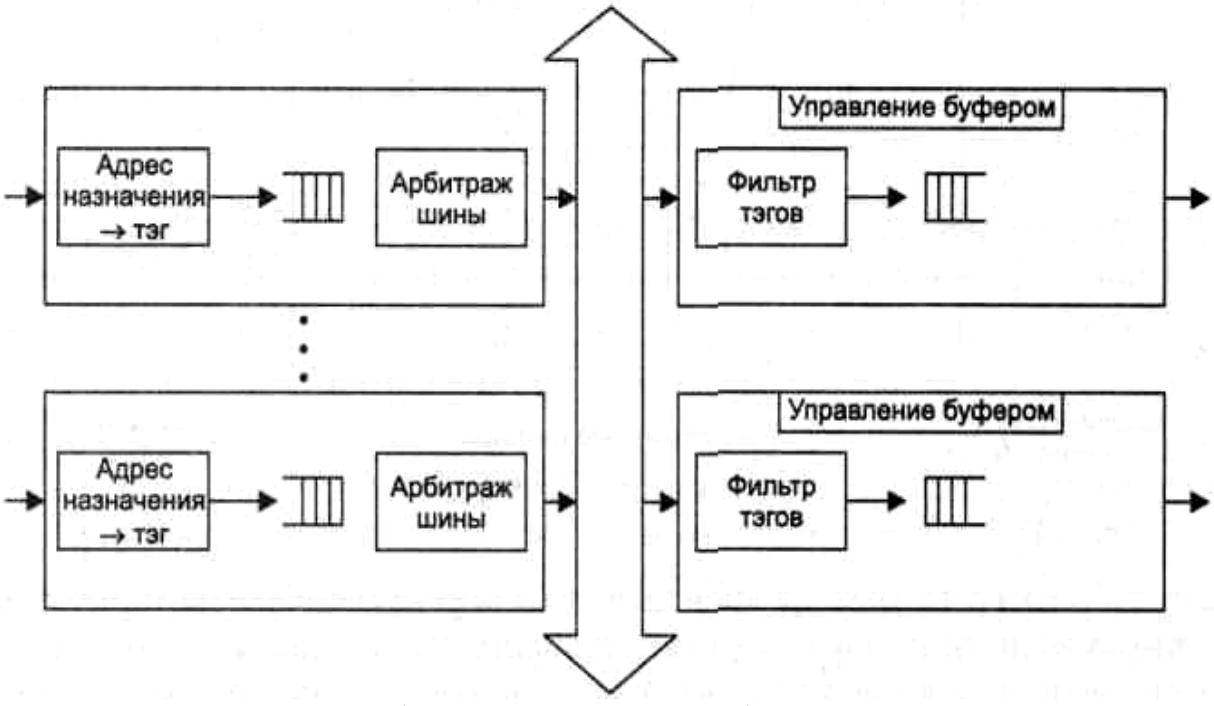


Рис. Архитектура коммутатора с общей шиной.

Каждый кадр передаваться по шине небольшими частями, по несколько байт (например, ячейками по 48 байт), чтобы обеспечить псевдопараллельную передачу кадров между несколькими портами. Входной блок процессора помещает в ячейку, переносимую по шине, тэг, в котором указывает номер порта назначения. Каждый выходной блок процессора порта содержит фильтр тэгов, который выбирает тэги, предназначенные данному порту. Достоинством коммутаторов с общей шиной является простота наращивания количества коммутируемых портов.

Коммутаторы с разделяемой памятью обеспечивают коммутацию портов при помощи общей разделяемой памяти (см. рис.).

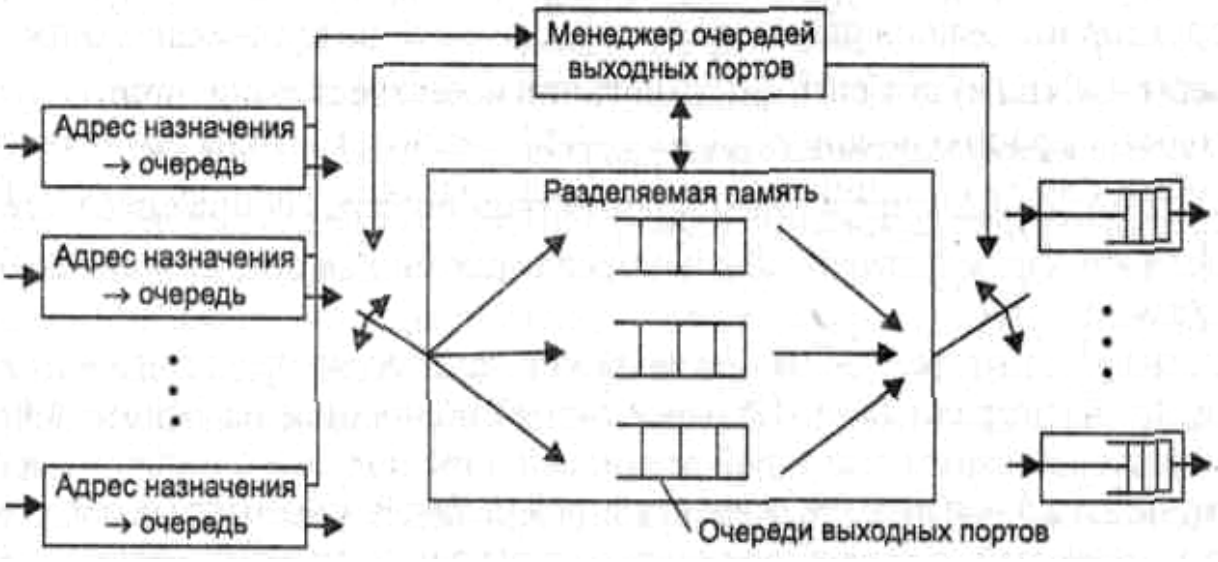


Рис. Архитектура коммутатора с общей разделяемой памятью.

Входные блоки процессоров портов соединяются с переключаемым входом разделяемой памяти, а выходные блоки этих же процессоров соединяются с переключаемым выходом этой памяти. Переключением входа и выхода разделяемой памяти управляет менеджер очередей выходных портов. В разделяемой памяти менеджер организует несколько очередей данных, по одной для каждого выходного порта. Входные блоки процессоров передают менеджеру портов запросы на запись данных в очередь того порта, который соответствует адресу назначения кадра. Менеджер по очереди подключает вход памяти к одному из входных блоков процессоров и тот переписывает часть данных кадра в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным блокам процессоров портов, и данные из очереди переписываются в выходной буфер процессора. Достоинством коммутаров с разделяемой памятью является гибкость и экономичность распределения общей памяти между отдельными портами, что снижает требования к размеру буферной памяти процессора каждого порта.

Комбинированные коммутаторы сочетают в себе достоинства различных типов архитектур. Пример такого коммутатора, сочетающего в себе скорость матричных коммутаторов и легкость наращивания числа портов коммутаторов с общей шиной, приведен на рис. .

Рис. Комбинированный коммутатор.

Коммутатор состоит из модулей с фиксированным количеством портов (2-12), выполненных в виде коммутационной матрицы. Модули соединены между собой при помощи общей шины. Если порты, между которыми нужно передать кадр данных, принадлежат одному модулю, то передача кадра осуществляется при помощи коммутационной матрицы. Если же порты принадлежат разным модулям, то процессоры общаются по общей шине.

Полнодуплексный и полудуплексный режим работы коммутатора, управление потоком кадров.

Обычно к коммутатору подключаются концентраторы, т.е. на отдельный порт подключается целый сегмент. Однако к порту могут подключаться и отдельные компьютеры (микросегментация). В таком случае, коммутатор и сетевая карта компьютера могут работать в полнодуплексном режиме, т.е. одновременно передавать

данные во встречных направлениях, увеличивая пропускную способность сети в два раза. Полнодуплексный режим возможен только если обе стороны - и сетевая карта и коммутатор - поддерживают этот режим. В полнодуплексном режиме не существует коллизий. Наложение двух кадров в кабеле считается нормальным явлением. Для выделения принимаемого сигнала, каждая из сторон вычитает из результирующего сигнала свой собственный сигнал.

При полудуплексном режиме работы, передача данных осуществляется только одной стороной, получающей доступ к разделяемой среде по алгоритму CSMA/CD. Полудуплексный режим фактически был подробно рассмотрен ранее.

При любом режиме работы коммутатора (полудуплексном или полнодуплексном) возникает проблема управления потоков кадров. Часто возникает ситуация, когда к одному из портов коммутатора подключен файл-сервер, к которому обращаются все остальные рабочие станции (см. рис.).

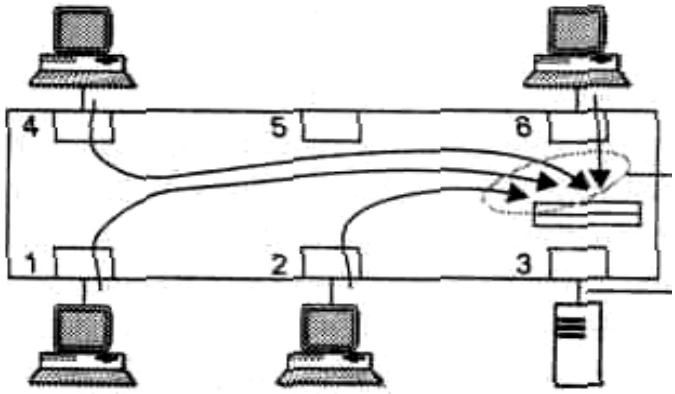


Рис. Отношение многие порты – к одному.

Если порт 3 работает на скорости 10 Мбит/с, а кадры с остальных четырех компьютеров поступают также со скоростью 10 Мбит/с, то не переданные кадры будут накапливаться в буфере порта 3 и, рано или поздно, этот буфер переполнится. Частичным решением данной проблемы было бы выделение для файл сервера порта 3, со скоростью 100 Мбит/с. Однако это не решает проблему, а лишь откладывает ее: со временем пользователи захотят более высоких скоростей работы сети, и коммутатор будет заменен на новый, у которого все порты будут работать на скорости 100 Мбит/с. Более продуманным решением, реализованном в большинстве коммутаторов, является управление потоком кадров, генерируемых компьютерами. В полнодуплексном режиме используются специальные служебные сигналы "Приостановить передачу" и "Возобновить передачу". Получив сигнал "Приостановить передачу" сетевая карта должна прекратить передачу кадров, вплоть до последующего сигнала "Возобновить передачу" (к сожалению в текущем стандарте 802.3x не предусмотрено частичное уменьшение интенсивности передачи кадров, возможен только полный запрет). В полудуплексном режиме используется "метод обратного давления" (backpressure) и "агрессивное поведение порта коммутатора". Оба метода позволяют реализовать достаточно тонкие механизмы управления потоком кадров, частично снижая их интенсивность, но не уменьшая ее до нуля.

Метод обратного давления (backpressure) состоит в создании искусственных коллизий в сегменте, который чересчур интенсивно посылает кадры в коммутатор. Для этого коммутатор обычно использует jam-последовательность (сигналы-помехи создающие и усиливающие коллизию), отправляемую на выход порта, к которому подключен сегмент (или компьютер), чтобы приостановить его активность.

Метод агрессивного поведения порта коммутатора основан на захвате среды либо после окончания передачи очередного пакета, либо после коллизии. В первом случае коммутатор оканчивает передачу очередного кадра и, вместо технологической паузы в 9,6 мкс, делает паузу в 9,1 мкс и начинает передачу нового кадра. Компьютер не сможет захватить среду, так как он выдержал стандартную паузу в 9,6 мкс и обнаружил после этого, что среда уже занята. Во втором случае кадры коммутатора и компьютера сталкиваются и фиксируется коллизия. Компьютер делает паузу после коллизии в 51,2 мкс, как это положено по стандарту, а коммутатор — 50 мкс. И в этом случае компьютеру не удается передать свой кадр. Коммутатор может пользоваться этим механизмом адаптивно, увеличивая степень своей агрессивности по мере необходимости.

Дополнительные возможности коммутаторов

Так как коммутатор представляет собой сложное вычислительное устройство, имеющее несколько процессорных модулей, то естественно нагрузить его помимо выполнения основной функции передачи кадров с порта на порт и некоторыми дополнительными функциями. Ниже описываются наиболее распространенные дополнительные функции коммутаторов.

1) Поддержка алгоритма Spanning Tree.

Как уже отмечалось, для нормальной работы коммутатора (моста) требуется отсутствие петлевых маршрутов в сети. Петлевые маршруты могут создаваться администратором специально, для образования резервных связей, или же возникать случайным образом, что вполне возможно, если сеть имеет сложную топологию связи и плохо структурирована или документирована. Алгоритм покрывающего дерева — Spanning Tree

Algorithm (STA) позволяет коммутаторам автоматически, при помощи обмена служебными пакетами, определять древовидную (без петель) конфигурацию связей в сети. В случае отказе какого-либо кабеля, порта или коммутатора, отказ обнаруживается автоматически, за счет постоянного тестирования связности сети служебными пакетами. После обнаружения потери связности протокол строит новое покрывающее дерево, если это возможно, и сеть автоматически восстанавливает работоспособность.

2) Трансляция протоколов канального уровня.

Коммутаторы позволяют преобразовывать кадры Ethernet в кадры FDDI, кадры Fast Ethernet в кадры Token Ring и т.п. Таким образом, если к одному порту коммутатора подсоединен сегмент FDDI, а к другому – сегмент Ethernet, то коммутатор позволит объединить эти две различные технологии канального уровня в единую сеть.

3) Фильтрация трафика.

Многие коммутаторы позволяют администраторам задавать дополнительные условия фильтрации кадров. Пользовательские фильтры предназначены для ограничения доступа определенных групп пользователей к определенным службам сети. Наиболее простыми являются пользовательские фильтры на основе MAC-адресов компьютеров. Самым простым вариантом является указание коммутатору отбрасывать кадры с определенным MAC-адресом. При этом пользователю, работающему на компьютере с данным MAC-адресом, полностью запрещается доступ к ресурсам другого сегмента сети. Часто администратору требуется задать более тонкие условия фильтрации, например запретить некоторому пользователю печатать свои документы на определенном сервере печати NetWare чужого сегмента, а остальные ресурсы этого сегмента сделать доступными. Для реализации такого фильтра нужно запретить передачу кадров с определенным MAC-адресом, в которых вложены пакеты IPX, в поле "номер сокета" которых будет указано значение, соответствующее службе печати NetWare. Коммутаторы не анализируют протоколы верхних уровней, поэтому администратору прейдется вручную, в шестнадцатеричной (двоичной) форме, задать такой фильтр и указать смещение и размер фильтра, относительно начала поля данных кадра канального уровня.

4) Приоритетная обработка кадров.

Использование коммутаторов позволяет реализовать приоритетную обработку кадров. Коммутатор обычно ведет для каждого входного и выходного порта не одну, а несколько очередей, причем каждая очередь имеет свой приоритет обработки. При этом коммутатор может быть сконфигурирован, например, так, чтобы передавать один низкоприоритетный пакет на каждые 10 высокоприоритетных пакетов. Основным вопросом при приоритетной обработке кадров является вопрос назначения кадру приоритета. Так как не все протоколы канального уровня поддерживают поле приоритета кадра (например, у кадров Ethernet оно отсутствует), то коммутатор должен использовать какой-либо дополнительный принцип для назначения приоритета кадру. Наиболее распространенный способ — приписывание приоритета портам коммутатора: кадр получает соответствующий приоритет в зависимости от того, через какой порт он поступил в коммутатор. Способ несложный, но недостаточно гибкий — если к порту коммутатора подключен не отдельный компьютер, а сегмент, то все узлы сегмента получают одинаковый приоритет. Более гибким способом является использование стандарта IEEE 802.1p: в кадре Ethernet, перед полем данных, предусматривается дополнительный заголовок, состоящий из двух байт, 3 бита из которых используются для указания приоритета кадра. При передаче кадра, компьютер, при помощи специального протокола, может запросить у коммутатора один из восьми уровней приоритета кадра. Установленный коммутатором приоритет, помещается в заголовок кадра и действует для всех коммутаторов в сети. При передаче кадра сетевой карте, не поддерживающей стандарт 802.1p, дополнительный заголовок, указывающий на приоритет кадра, должен быть удален.

5) Виртуальные локальные сети (Virtual LAN, VLAN).

Коммутаторы позволяют реализовывать технологии виртуальных локальных сетей. Несмотря на схожесть терминов, не следует путать виртуальные частные сети (VPN – Virtual Private Network) и виртуальные локальные сети (Virtual LAN, VLAN). Виртуальные частные сети позволяют на сетевом уровне безопасно объединить через глобальные сети (например, Internet) или линии телефонной связи несколько локальных сетей, в единую виртуальную ЛВС. Виртуальные локальные сети позволяют на канальном уровне выделить внутри одной, физически существующей ЛВС, несколько изолированных друг от друга виртуальных ЛВС.

Виртуальной сетью называется группа узлов сети, кадры которых, в том числе и широковещательные, на канальном уровне полностью изолированы от других узлов сети (см. рис.).

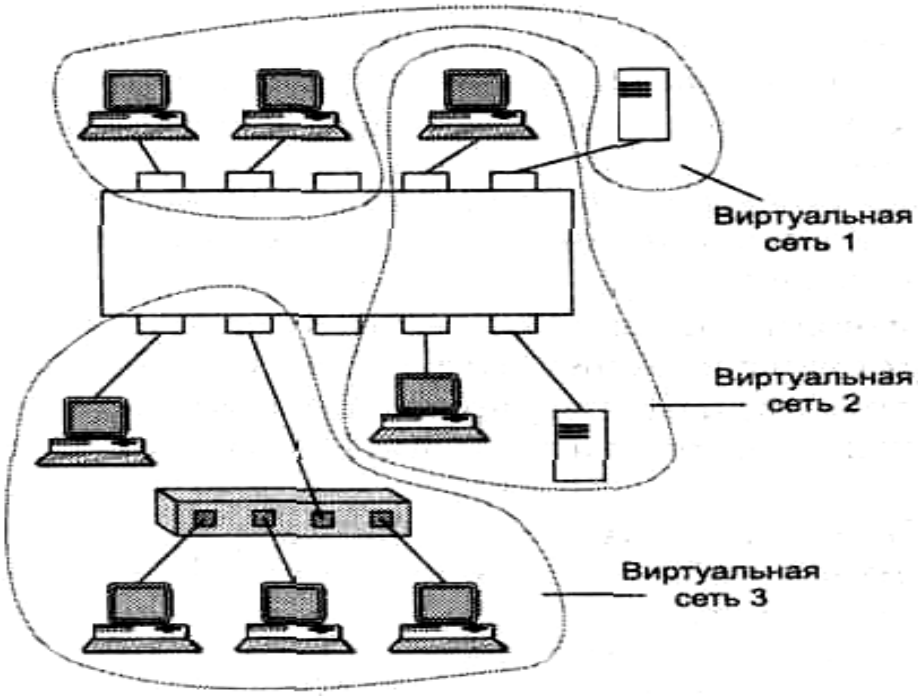


Рис. Виртуальная локальная сеть

Несмотря на то, что все узлы сети могут, например, быть подключены к одному и тому же коммутатору, передача кадров на канальном уровне между ними невозможна. Передача кадров между разными виртуальными сетями возможно только на сетевом уровне, при помощи маршрутизатора. В то же время, внутри виртуальной сети кадры передаются коммутатором стандартным образом, на канальном уровне и только на тот порт, который необходимо. Виртуальные сети могут пересекаться, если один или несколько компьютеров входят в состав более чем одной виртуальной сети (см. рис.). В таком случае, виртуальные локальные сети могут взаимодействовать между собой через эти общие компьютеры, которые могут

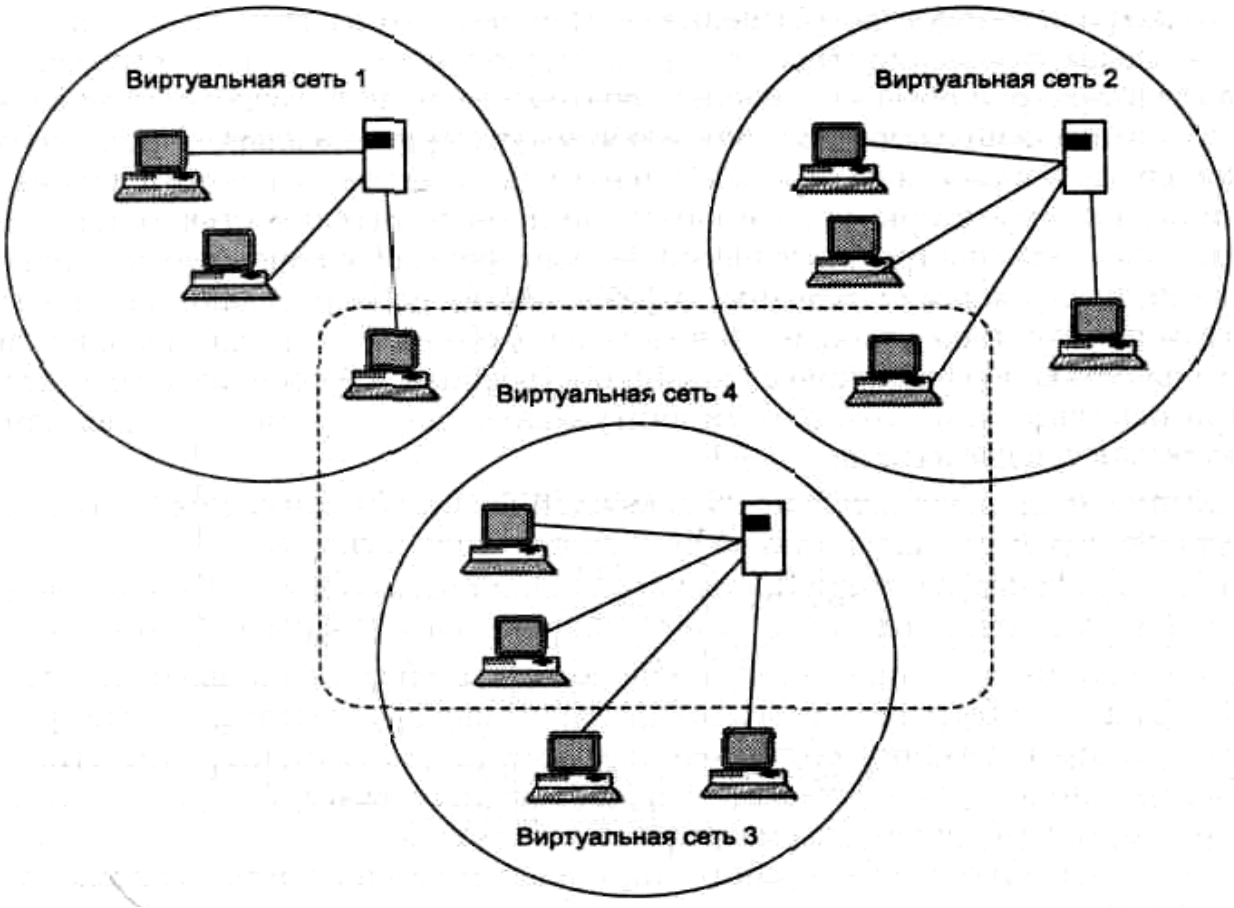


Рис. Пересечение виртуальных локальных сетей.

Технология виртуальных сетей создает гибкую основу для построения крупной сети, соединенной маршрутизаторами, так как коммутаторы позволяют создавать полностью изолированные сегменты программным путем, что очень удобно в крупных сетях. До появления технологии VLAN, для создания отдельной сети использовались либо физически изолированные сегменты коаксиального кабеля, либо несвязанные между собой сегменты, построенные на повторителях и мостах. Затем эти сети связывались маршрутизаторами в единую составную сеть (см. рис.).

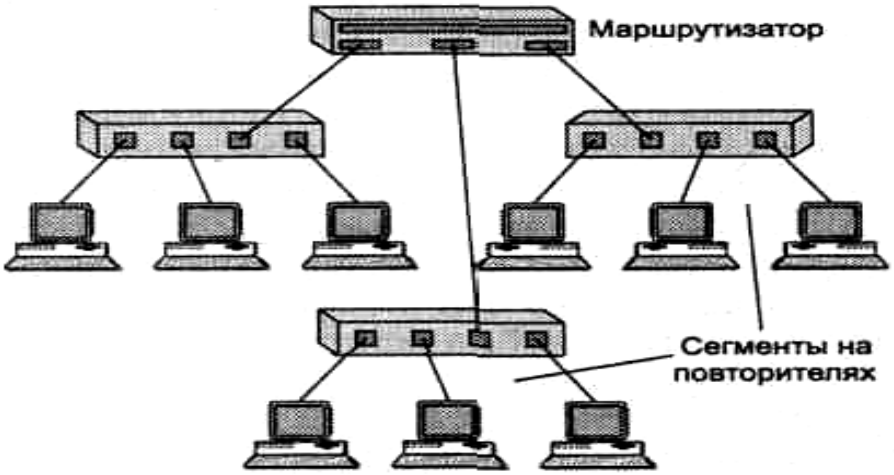


Рис. Сеть, состоящая из физически независимых подсетей.

Любое изменение структуры такой сети означало физические изменения в подключении того или иного оборудования к портам концентраторов, коммутаторов и маршрутизаторов, изменения в прокладке кабеля и т.д. В больших сетях это требовало значительных объемов работ, что повышало вероятность ошибок. При создании виртуальных сетей программным способом, порты коммутатора при помощи графической программы легко группируются в отдельные виртуальные сети. Другим, более гибким способом, является группировка в виртуальные сети не портов коммутатора, а MAC-адресов отдельных компьютеров.

Характеристики, влияющие на производительность коммутаторов

При выборе коммутатора следует в первую очередь обращать внимание на характеристики, обеспечивающие его производительность, т.к. именно это свойство послужило причиной вытеснения мостов коммутаторами. Ниже приведены некоторые характеристики:

1) Скорость фильтрации/продвижения кадров (кадров в секунду), пропускная способность (мегабит в секунду), задержка передачи кадра.

Коммутатор является *неблокирующим*, если он может передавать кадры через свои порты с той же скоростью, с какой они на них поступают. Необходимо учитывать для кадров какого протокола и какой длины указана пропускная способность. Максимальная пропускная способность всегда достигается на кадрах максимальной длины, так как при этом доля накладных расходов на служебную информацию кадра гораздо ниже, чем для кадров минимальной длины, а время обработки кадра (в расчете на один байт полезной информации), существенно меньше. Поэтому коммутатор может быть блокирующим для кадров минимальной длины, но при этом иметь очень хорошие показатели пропускной способности. Коммутатор – это многопортовое устройство, поэтому для него все приведенные выше характеристики (кроме задержки передачи кадра) можно давать в общей сумме, или в расчете на один порт. Обычно производители коммутаторов указывают общую максимальную пропускную способность устройства.

2) Тип коммутации — "на лету" или с полной буферизацией.

При коммутации на лету передача кадра начинается сразу после приема первых нескольких байт заголовка. При коммутации с полной буферизацией кадр должен быть полностью принят в буферную память до начала передачи. Разницу между этими характеристиками коммутаторов иллюстрирует следующая таблица:

Таблица

Возможности коммутаторов при коммутации "на лету" и с полной буферизацией.

Функция	Налету	С буферизацией
Защита от плохих кадров	Нет	Да
Трансляция протоколов разнородных сетей (Ethernet Token Ring, FDDI, ATM)	Нет	Да
Задержка передачи пакетов	Низкая (5-40 мкс) при низкой нагрузке, средняя при высокой нагрузке	Средняя при любой нагрузке
Поддержка резервных связей	Нет	Да
Функция анализа трафика	Нет	Да

Так как каждый способ имеет свои достоинства и недостатки, в тех моделях коммутаторов, которым не нужно транслировать протоколы, иногда применяется механизм адаптивной смены режима работы

коммутатора. Основной режим такого коммутатора — коммутация "на лету", но коммутатор постоянно контролирует трафик и при превышении интенсивности появления плохих кадров некоторого порога переходит на режим полной буферизации. Затем коммутатор может вернуться к коммутации "на лету".

3) Размер адресной таблицы.

Максимальная емкость адресной таблицы определяет предельное количество MAC-адресов, с которыми может одновременно оперировать коммутатор. Так как коммутаторы чаще всего используют для выполнения операций каждого порта выделенный процессорный блок со своей памятью для хранения экземпляра адресной таблицы, то размер адресной таблицы для коммутаторов обычно приводится в расчете на один порт. Каждый порт хранит только те наборы адресов, с которыми он работал в последнее время. Недостаточная емкость адресной таблицы может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если адресная таблица процессора порта полностью заполнена, а он встречает новый адрес источника в поступившем пакете, процессор должен вытеснить из таблицы какой-либо старый адрес и поместить на его место новый. Эта операция сама по себе отнимет у процессора часть времени, но главные потери производительности будут наблюдаться при поступлении кадра с адресом назначения, который пришлось удалить из адресной таблицы. Так как адрес назначения кадра неизвестен, то коммутатор должен передать этот кадр на все остальные порты. Эта операция будет создавать лишнюю работу для многих процессоров портов, кроме того, копии этого кадра будут попадать и на те сегменты сети, где они совсем не обязательны. Некоторые производители коммутаторов решают эту проблему за счет того, что один из портов коммутатора конфигурируется как магистральный порт, на который по умолчанию передаются все кадры с неизвестным адресом. Передача кадра на магистральный порт производится в расчете на то, что этот порт подключен к вышестоящему коммутатору, который имеет большую емкость адресной таблицы и знает, куда нужно передать любой кадр.

4) Объем буфера кадров.

Внутренняя буферная память коммутатора нужна для временного хранения кадров данных в тех случаях, когда их невозможно немедленно передать на выходной порт. Буфер предназначен для сглаживания кратковременных пульсаций нагрузки на сеть. Каждый процессорный модуль порта обычно имеет свою буферную память. Чем больше объем этой памяти, тем менее вероятны потери кадров при перегрузках. Обычно коммутаторы, предназначенные для работы в ответственных частях сети, имеют буферную память в несколько десятков или сотен килобайт на порт. Хорошо, когда эту буферную память можно перераспределять между несколькими портами, так как одновременные перегрузки по нескольким портам маловероятны. Дополнительным средством защиты может служить общий для всех портов буфер в модуле управления коммутатором. Такой буфер обычно имеет объем в несколько мегабайт.

5) Производительность процессоров портов, производительность внутренней шины коммутатора.

Необходимо следить, чтобы производительность общей шины (при архитектуре с общей шиной) или производительность процессоров портов не стала "узким местом" в коммутаторе.

3.4. Маршрутизатор (router)

Маршрутизаторы необходимы в крупных сетях, для объединения сегментов, построенных на концентраторах, мостах и коммутаторах. Маршрутизатор может быть реализован в виде отдельного высокопроизводительного устройства (например, маршрутизаторы компании Cisco Systems), или функцию маршрутизации может выполнять сетевая операционная система обычного компьютера, подключенного одновременно к нескольким сетям, при помощи нескольких сетевых карт (шлюз). Маршрутизаторы работают на сетевом уровне модели OSI и не накладывают ограничений на топологию сети. Если для мостов и коммутаторов обязательно отсутствие петлевых маршрутов в сети (древовидная структура), то маршрутизатор работает в сетях с произвольной топологией и обеспечивает выбор оптимального маршрута для доставки пакетов. Использование древовидной структуры для крупных сетей нерационально, т.к. в таком случае на корневой коммутатор (мост) приходится слишком большая нагрузка, а его отказ приводит к распадению сети на отдельные фрагменты и потере пользователями доступа к большому количеству ресурсов сети. Поэтому рационально строить сети по децентрализованному принципу, когда между любыми двумя компьютерами может существовать множество маршрутов. Именно нахождением и ведением таблицы таких маршрутов (таблицы маршрутизации) и доставкой пакетов по оптимальному маршруту занимается маршрутизатор.

Другой функцией маршрутизаторов является объединение в единую сеть сегментов, работающих на различных протоколах канального уровня. Например, объединение сегментов Fast Ethernet и FDDI. Маршрутизатор работает на сетевом уровне модели OSI (например, по протоколу IP), и для него не существенно какие протоколы канального уровня используются в сегментах. Трансляция протоколов (кадры Fast Ethernet в кадры FDDI) может осуществляться и некоторыми моделями коммутаторов, однако такая возможность появилась сравнительно недавно, и исторически для объединения разнородных сетей используют маршрутизаторы. Кроме того, коммутаторы в некоторых случаях не могут корректно выполнить трансляцию кадров. Например, коммутаторами не поддерживается функция фрагментации кадров и, если в объединяемых сетях не совпадают максимально допустимые размеры кадров, то коммутатор не сможет транслировать очень большие кадры.

Сегодня считается, что любая крупная сеть должна включать изолированные сегменты, соединенные маршрутизаторами, иначе потоки ошибочных кадров, например широковещательных, будут периодически затапливать всю сеть через прозрачные для них коммутаторы (мосты), приводя ее в неработоспособное состояние. Кроме того, использование маршрутизаторов позволяет структурировать сеть (подсеть отдела кадров, подсеть бухгалтерии и т.п.) и легче реализовывать политику безопасности, за счет использования межсетевых экранов. Межсетевой экран (firewall, брандмауэр) – это специальное программное обеспечение, которое установлено на маршрутизаторе, или компьютере-шлюзе, выполняющем функции маршрутизатора, и позволяющее контролировать доступ пользователей к тем или иным ресурсам сети. Для межсетевого экрана задаются правила фильтрации вида: "через межсетевой экран допускается прохождение пакетов с IP-адресом отправителя 172.18.10.1 (порт 80) и IP-адресом получателя 192.168.1.1 (порт 21), в четверг с 15.00 до 19.00". Пакеты, не удовлетворяющие правилам фильтрации отбрасываются, а факт их наличия регистрируется в специальном журнале.

В крупных сетях выбор наилучшего маршрута часто является достаточно сложной задачей, с математической точки зрения. Особенно интенсивных вычислений требуют протоколы OSPF, NLSP, IS-IS, вычисляющие оптимальный путь на графе. Кроме того маршрутизатор вынужден выполнять буферизацию, фильтрацию, фрагментацию пакетов и другие задачи. При этом очень важна производительность маршрутизатора, поэтому типичный маршрутизатор крупных сетей является мощным вычислительным устройством с одним или даже несколькими процессорами (часто специализированными или построенными на RISC-архитектуре) и сложным программным обеспечением, работающим под управлением специализированной операционной системы реального времени. Многие разработчики маршрутизаторов построили в свое время такие операционные системы на базе ОС Unix, естественно, значительно ее переработав.

Алгоритмы маршрутизации

Маршрутизация – это выбор маршрута доставки пакета. Частично, алгоритм маршрутизации уже был рассмотрен в лекции, посвященной протоколу IP. Здесь будет рассмотрена только общая классификация алгоритмов маршрутизации, не упоминавшихся ранее.

По степени распределенности схемы маршрутизации выделяют следующие алгоритмы маршрутизации:

- Одношаговые алгоритмы. Наиболее широко распространены в сетях. В таблице маршрутизации хранится информация только об одном шаге маршрута (ближайший маршрутизатор на пути к адресату). При отсутствии возможности доставки пакета напрямую (когда маршрутизатор различными сетевыми интерфейсами одновременно подключен и к сети отправителя и к сети адресата), пакет доставляется на следующий ближайший маршрутизатор на пути к адресату, который анализирует свои таблицы маршрутизации и занимается дальнейшей доставкой пакета. Подробнее см. лекции по протоколу IP.
- Многошаговые алгоритмы. В таблице маршрутизации указываются все шаги маршрута (промежуточные маршрутизаторы), которые должен пройти пакет. Схема работы - аналогично мостам, с маршрутизацией от источника (см. ранее). В сетях распространена мало. Однако в новой версии протокола IP (IPv6), наряду с классической одношаговой маршрутизацией, будет разрешена и маршрутизация от источника.

По способу построения таблиц маршрутизации выделяют следующие алгоритмы маршрутизации:

- Алгоритмы статической маршрутизации. Все записи в таблице маршрутизации задаются администратором вручную. Пригоден только для небольших сетей. В крупных сетях применяется только совместно с алгоритмом динамической маршрутизации.
- Алгоритмы динамической маршрутизации (таблицы маршрутизации составляются и обновляются автоматически, на основании имеющейся информации о непосредственно подключенных к маршрутизатору сетях и информации от соседних маршрутизаторов, передаваемой по протоколам RIP, OSPF, NLSP, подробнее см. ниже).
- Алгоритмы простой маршрутизации. В сетях практически не применяются. Используется случайная маршрутизация (прибывший пакет посылается в первом попавшемся случайном направлении, кроме исходного), лавинная маршрутизация (пакет широковещательно посылается по всем возможным направлениям, кроме исходного), маршрутизация по предыдущему опыту (выбор маршрута осуществляется аналогично выбору маршрута в прозрачных мостах).

По использованию маски подсети в процессе маршрутизации IP-пакетов выделяют:

- Маршрутизацию на основании классов IP-адресов, без использования маски подсети.
- Бесклассовая междоменная маршрутизация, с использованием маски подсети.

При маршрутизации на основании классов IP-адресов, в таблицах маршрутизации маски подсетей не хранятся. Решение о том, является ли данный IP-адрес адресом сети или адресом конкретного компьютера принимается на основании класса IP-адреса (у сетей класса C адрес компьютера находится в последнем октете, у сетей класса B адрес компьютера находится в последних двух октетах и т.д.). Такой подход прост, но создает неудобства, т.к. минимальный размер подсети составляет 253 компьютера (сеть класса C), что

является нерациональным расходованием адресов и не позволяет структуризировать сеть на более мелкие подсети. Поэтому постепенно в сетях происходит переход на маршрутизацию с использованием масок подсети - бесклассовая междоменная маршрутизация (CIDR, Classless Inter-Domain Routing). При этом подходе подсетям выделяются непрерывные диапазоны IP-адресов так, чтобы номер компьютера и номер сети можно было описать при помощи маски подсети (подробнее см. лекции по протоколу IP). При обмене информацией между маршрутизаторами (например, по протоколу RIPv2), вместе с информацией о маршрутах передается и информация о масках подсетей для соответствующих IP-адресов.

Протоколы динамической маршрутизации.

Протоколы маршрутизации обеспечивают обмен служебной информацией, необходимой для построения таблиц маршрутизации. Существует множество протоколов маршрутизации, однако здесь мы рассмотрим только два протокола: RIP (представитель дистанционно-векторных протоколов) и OSPF (представитель протоколов состояния связей).

Построение таблицы маршрутизации по протоколу RIP (Routing Information Protocol) происходит следующим образом:

1) Маршрутизаторы создают минимальные таблицы маршрутизации, на основании имеющейся информации о сетях, непосредственно подключенным к их сетевым интерфейсам.
2) Маршрутизаторы рассылают минимальные таблицы соседям (соседями считаются те маршрутизаторы, которые могут получить сообщение напрямую, не пользуясь услугами промежуточных маршрутизаторов, т.е. маршрутизаторы которые одним из своих сетевых интерфейсов подключены к той же сети, что и маршрутизатор, отправляющий сообщение).
3) Маршрутизаторы анализируют полученные минимальные таблицы других маршрутизаторов, наращивая поле "метрика" (расстояние до сети/компьютера) на единицу, учитывая таким образом тот соседний маршрутизатор, от которого была получена минимальная таблица, как еще один маршрутизатор на пути до сети назначения. После этого полученная минимальная таблица сравнивается с уже имеющейся таблицей маршрутизации. Если в обоих таблицах имеется несколько маршрутов до одной и той же сети, то в результирующую таблицу попадает вариант с наименьшей метрикой (расстоянием до сети).
4) Рассылка новой, уже не минимальной, таблицы соседям и последующая обработка полученных от соседей не минимальных таблиц. Соседям рассылается полный вариант имеющейся таблицы маршрутизации, за исключением информации о сетях, которая была получена непосредственно от самих этих соседей. Делается это для предотвращения создания петлевых маршрутов и зацикливания пакетов (техника "расщепления горизонта" – split horizont). Этап 4 повторяется циклически каждые 30 секунд (протокол RIP IP). В результате, все маршрутизаторы в сети будут иметь корректную таблицу маршрутизации: информация из таблиц любого маршрутизатора, через соседей, рано или поздно дойдет до любого другого маршрутизатора в сети. Более того, при изменениях в сети (подключение к какому-либо маршрутизатору новой сети или временная недоступность старой сети) информация об изменениях также распространится по сети.
5) Для автоматического обновления таблиц маршрутизации, каждая запись, созданная при помощи протокола RIP, имеет свой срок жизни (TTL, Time To Live). В RIP IP срок жизни записи равен шести периодам рассылки таблиц маршрутизации, т.е. 180 секунд. Если какой-либо маршрутизатор выходит из строя 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 и т.д. Как видно из объяснения, сведения о недоступных через отказавший маршрутизатор сетях распространяются по сети не очень быстро. Поэтому есть более быстрый способ объявить сеть недоступной. Если отказал не весь маршрутизатор (что случается редко), а только один из его интерфейсов, подключенных к какой-либо сети, то при следующем обмене таблицами маршрутизации (через 30 секунд), маршрутизатор укажет "бесконечное" расстояние до недоступной сети, что приведет к исключению данного маршрута из таблиц других маршрутизаторов. Бесконечному расстоянию в протоколе RIP IP соответствует метрика 16 (16 маршрутизаторов между отправителем и получателем). Такое небольшое значение "бесконечного" расстояния связано с низкой скоростью распространения информации об отказах между маршрутизаторами (см. выше). Большее значение может привести к длительным периодам зацикливания и потерь пакетов.

Протокол OSPF (Open Shortest Path First) является более современным и эффективным, чем протокол RIP.

В OSPF процесс построения таблицы маршрутизации происходит по следующим этапам:

1) 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.

2) 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 математически 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.

3) После первоначального построения таблицы маршрутизации необходимо отслеживать изменения состояния сети и вносить коррективы в таблицу маршрутизации. Для контроля состояния связей в сети OSPF-маршрутизаторы не используют обмен полной таблицей маршрутизации, как это не очень рационально делают RIP-маршрутизаторы. Вместо этого они передают специальные короткие сообщения HELLO. Если состояние сети не меняется, то OSPF-маршрутизаторы 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 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).

Автономные системы, протоколы внешней маршрутизации.

Протоколы RIP и OSPF используются внутри локальных сетей, или, если рассматривать сеть Internet, внутри автономных систем.

Автономная система – это объединение локальных сетей с одинаковой маршрутной политикой и общей администрацией, например совокупность сетей компании Ростелеком. Каждая автономная система регистрируется (за довольно существенную плату) в региональной регистратуре Internet. Для Европы и России – это RIPE (http://www.ripe.net). Если Вам известен IP-адрес, то используя программу ***WhoIS*** (входит во все дистрибутивы Linux/Unix, или, например, в программу IPTOOLS для Windows) можно обратиться в одну из региональных регистратур (whois.ripe.net, whois.internic.net, whois.ripn.net, whois.arin.net, whois.apnic.net, whois.nic.mil, whois.nic.gov) и получить сведения об автономной системе, за которой числится данный IP-адрес: адрес организации, имя ответственного, телефоны и т.д.

Маршрутизация между автономными системами осуществляется пограничными маршрутизаторами (border gateways). При маршрутизации используются протоколы внешней маршрутизации, в частности BGP (Border Gateway Protocol). Его принципиальным отличием от протоколов внутренней маршрутизации (RIP, OSPF) является наличие маршрутной политики. Маршрутная политика позволяет передавать другим погарничным маршрутизаторам не все существующие маршруты, а только те, которые администрация автономной системы сочтет нужными. Также, для различных маршрутов можно задавать дополнительные параметры, характеризующие пропускную способность маршрута, стоимость транзита трафика по данному маршруту и т.д. Приведем пример (см. рис.).

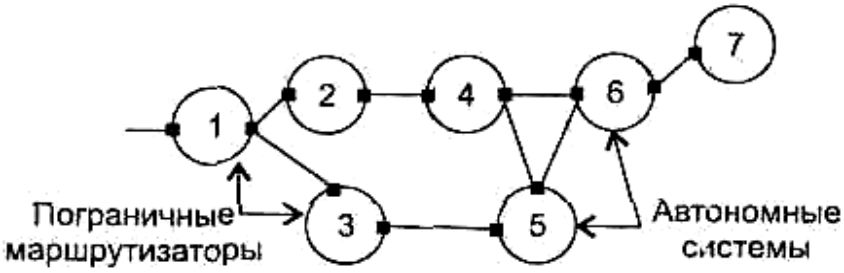


рис. Использование протокола BGP.

Пример 1

Администрация автономной системы 2 (АС2) не хочет, чтобы сетевой трафик из автономной системы 1 (АС1) проходил транзитом через нее. Поэтому, хотя автономные системы АС4, АС5, АС6, АС7, АС3 и доступны для АС1 через АС2, однако АС2 не объявляеет об этом. Такое возможно только при ручном составлении таблиц маршрутизации или использовании протокола BGP (протоколы RIP и OSPF не позволяют этого сделать).

Пример 2

Кратчайший маршрут из АС1 в АС5 лежит через АС3. Однако стоимость транзита через АС3 для АС1 чрезвычайно велика. Другой маршрут АС1–АС2–АС4–АС5 короче маршрута АС1–АС2–АС4–АС6–АС5, однако связь АС4-АС5 значительно "медленне" (33,6 Кбит/с), чем связь АС4–АС6–АС5 (2,488 Гбит/с),

поэтому маршрут АС1–АС2–АС4–АС6–АС5 предпочтительнее. Протокол BGP, при соответствующей настройке пограничных маршрутизаторов, позволяеет учесть все эти особенности.

Классификация маршрутизаторов.

По областям применения маршрутизаторы делятся на несколько классов.

Магистральные маршрутизаторы (backbone routers) предназначены для построения главной магистрали сети. Это наиболее мощные устройства, способные обрабатывать несколько сотен тысяч или даже несколько миллионов пакетов в секунду, имеющие большое количество интерфейсов локальных (Ethernet, Token Ring, FDDI) и глобальных сетей (T1/E1, SDH, ATM). Чаще всего магистральный маршрутизатор выполнен по модульной схеме на основе шасси, с большим количеством слотов (≈12). Большое внимание уделяется надежности и отказоустойчивости маршрутизатора, которая достигается за счет системы терморегуляции, избыточных источников питания, заменяемых "на ходу" модулей. 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. Такой маршрутизатор обычно представляет собой упрощенную версию магистрального маршрутизатора. Если он выполнен на основе шасси, то количество слотов его шасси меньше (≈4-5). Возможна также реализация с фиксированным числом портов. Меньше список поддерживаемых интерфейсов локальных и глобальных сетей. Примерами маршрутизаторов региональных отделений могут служить маршрутизаторы BLN, ASN компании Nortel Networks, Cisco 3600, Cisco 2500, NetBuilder II компании 3Com.

Маршрутизаторы удаленных офисов соединяют, как правило, единственную локальную сеть удаленного офиса с центральной сетью предприятия по глобальной связи. Обычно такие маршрутизаторы поддерживают для локальной сети интерфейс Ethernet 10 Мбит/с (Fast Ethernet 100 Мбит/с), а для глобальной сети — выделенную линия со скоростью 64 Кбит/с, 1544 Кбит/с или 2 Мбит/с (может также поддерживаться коммутируемая телефонная линия, в качестве резервной связи). 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. В коммутаторах 3-го уровня отсутствуют низкоскоростные порты (такие как модемные порты 33,6 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. Подробнее коммутаторы третьего уровня будут рассмотрены ниже.

Дополнительные функции маршрутизаторов.

1) Поддержка нескольких сетевых протоколов. Приоритеты сетевых протоколов.

Маршрутизатор, поддерживающий несколько протоколов сетевого уровня (например, IP и IPX), называют многопротокольным. 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.

2) Поддержка одновременно нескольких протоколов маршрутизации.

В протоколах маршрутизации обычно предполагается, что маршрутизатор автоматически строит свою таблицу на основе работы только этого одного протокола маршрутизации (например, только протокола RIP). Тем не менее, иногда в большой сети приходится поддерживать одновременно несколько протоколов маршрутизации, 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 (OSPF или NLSP), а низшим приоритетов обладают маршруты дистанционно-векторных протоколов (RIP), как самых несовершенных.

3) 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 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. Разработчики маршрутизаторов исправляют недостатки протоколов RIP, OSPF и NLSP, вводя в маршрутизаторы поддержку правил передачи и использования маршрутной информации, подобных тем, которые рекомендует BGP.

4) Поддержка немаршрутизируемых протоколов.

Немаршрутизируемые протоколы, такие как NetBIOS (NetBEUI), не работают с адресами сетевого уровня. Маршрутизаторы могут обрабатывать пакеты таких протоколов двумя способами. В первом случае они работают с пакетами этих протоколов на канальном уровне, как прозрачные мосты. Маршрутизатор необходимо сконфигурировать так, чтобы по отношению к немаршрутизируемым протоколам на некоторых портах он выполнял функции моста, а по отношению к маршрутизируемым протоколам — функции маршрутизатора. Такой маршрутизатор иногда называют brouter (bridge + router). Другим способом передачи пакетов немаршрутизируемых протоколов является инкапсуляция этих пакетов в пакеты протокола сетевого уровня, чаще всего в IP-пакеты. Некоторые производители маршрутизаторов разработали собственные протоколы, специально предназначенные для инкапсуляции немаршрутизируемых пакетов.

5) 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 (RIP, OSPF, NLSP) и вычислении оптимального пути к каждой целевой сети по некоторому критерию. Для вычисления оптимального пути на графе, как того требуют протоколы OSPF и NLSP, необходимы значительные вычислительные мощности. После того как таблица маршрутизации составлена, продвижение пакетов происходит весьма просто, на основании просмотра таблицы маршрутизации. Некоторые маршрутизаторы поддерживают только функции продвижения пакетов по готовой таблице маршрутизации. Такие маршрутизаторы являются усеченными маршрутизаторами, так как для их полноценной работы требуется наличие обычного маршрутизатора, у которого можно взять готовую таблицу маршрутизации. Этот маршрутизатор часто называется сервером маршрутов. Отказ от самостоятельного 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.

Основные технические характеристики маршрутизатора.

1) Перечень поддерживаемых сетевых протоколов и протоколов машрутизации.

При выборе маршрутизатора необходимо учитывать какие протоколы (в том числе и устаревшие) используются или будут использоваться на предприятии. Перечень поддерживаемых сетевых протоколов обычно включает протоколы IP, IPX, AppleTalk, CONS и CLNS OSI, 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.

2) Перечень поддерживаемых интерфейсов локальных и глобальных сетей.

Маршрутизатор должен иметь интерфейсы ко всем протоколам канального уровня, использующимся в локальной сети предприятия, а также для связи с глобальными сетями. Для локальных сетей – это интерфейсы Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN. Для глобальных связей – это интерфейсы V.21-V.90 (модем), T1/E1, T3/E3, SONET/SDH, ISDN, интерфейсы к сетям X.25, Frame Relay, ATM, а также поддержка протокола канального уровня PPP.

3) Общая производительность маршрутизатора.

Общая производительность маршрутизаторов колеблется от нескольких десятков тысяч пакетов в секунду до нескольких миллионов пакетов в секунду и зависит от многих факторов, наиболее важными из которых являются: тип используемых процессоров, эффективность программной реализации протоколов, архитектурная организация вычислительных и интерфейсных модулей. Наиболее производительные маршрутизаторы имеют мультипроцессорную архитектуру: несколько мощных центральных процессоров 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.

3.5. Корпоративные модульные концентраторы.

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.

Основная идея разработчиков таких устройств заключается в создании программно настраиваемой конфигурации сети. 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 (см. рис.). Ввиду того, что отказ корпоративного модульного концентратора приводит к очень тяжелым последствиям, в их конструкцию вносится большое количество средств обеспечения отказоустойчивости.

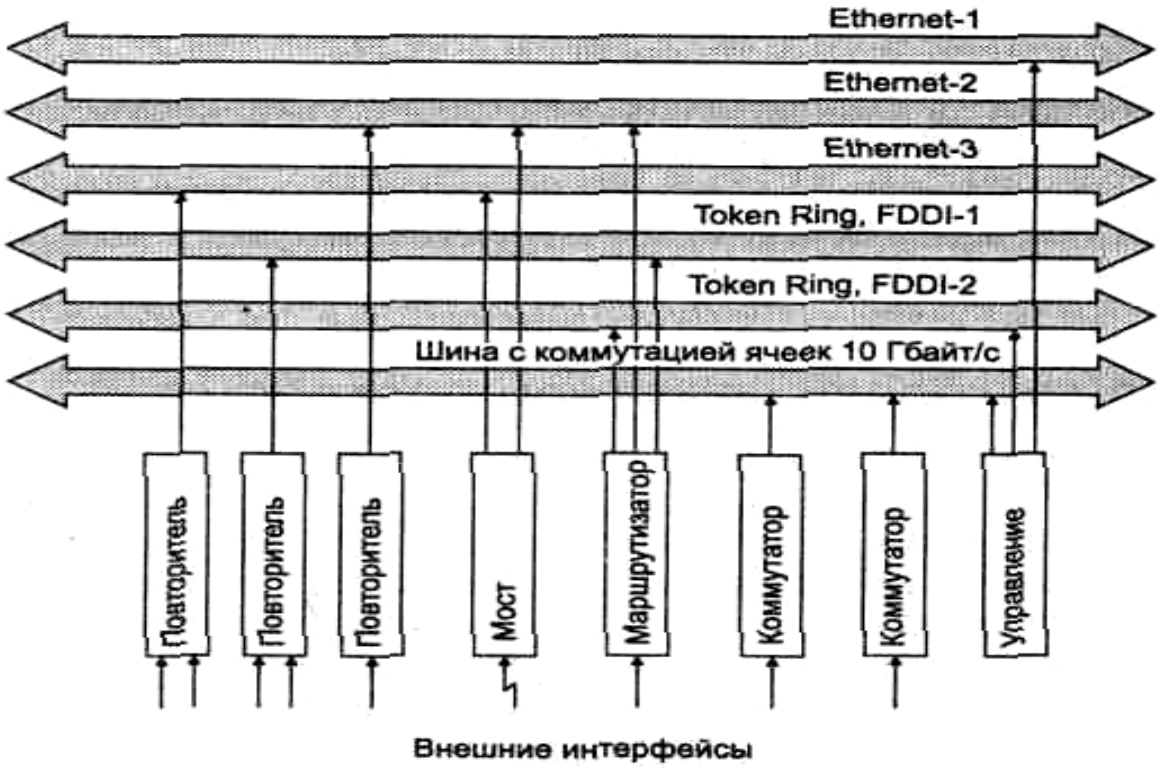


Рис. Структура корпоративного модульного концентратора

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

3.6. Коммутаторы третьего уровня.

В последние время наметилась тенденция сглаживания различий между коммутаторами и маршрутизаторами. Производители коммутаторов наделяют некоторые свои модели функциями маршрутизации, что позволяет использовать скорость коммутаторов и преимущества маршрутизаторов. Такие коммутаторы получили название коммутаторов третьего уровня. Функции коммутации и маршрутизации могут быть совмещены двумя способами:

1) Классическим, когда маршрутизация выполняется по каждому пакету, требующему передачи из сети в сеть, а коммутация выполняется для пакетов, принадлежащих одной сети.

2) Методом маршрутизации потока, когда маршрутизируется несколько первых пакетов устойчивого потока, а все остальные пакеты этого потока коммутируются.

Коммутаторы 3-го уровня с классической маршрутизацией.

Обычный коммутатор "прозрачен" для компьютеров сети, не имеет собственных MAC-адресов портов и захватывает все кадры, приходящие на порт, независимо от их адреса назначения, для последующей коммутации. Классический коммутатор 3-го уровня, подобно обычному коммутатору, захватывает все кадры своими портами независимо от их MAC-адресов, однако порты коммутатора 3-го уровня имеют и собственные MAC-адреса. Если захваченный кадр направлен на MAC-адрес какого-либо компьютера в сети, то пакет коммутируется. Если захваченный кадр направлен на MAC-адрес порта коммутатора, то пакет маршрутизируется. 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. Примерами таких коммутаторов могут служить хорошо известные коммутаторы LANplex (теперь CoreBuilder) 6000 и 2500 компании 3Com.

Коммутаторы 3-го уровня с маршрутизацией 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. Последующие пакеты обрабатываются по схеме коммутации. 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.

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 (QoS, Quality of Service), т.е. одинаковые требования к скорости передачи данных, задержках в передаче пакетов, доле потерь пакетов и т.п. Приведем пример использования потоков для ускорения маршрутизации.

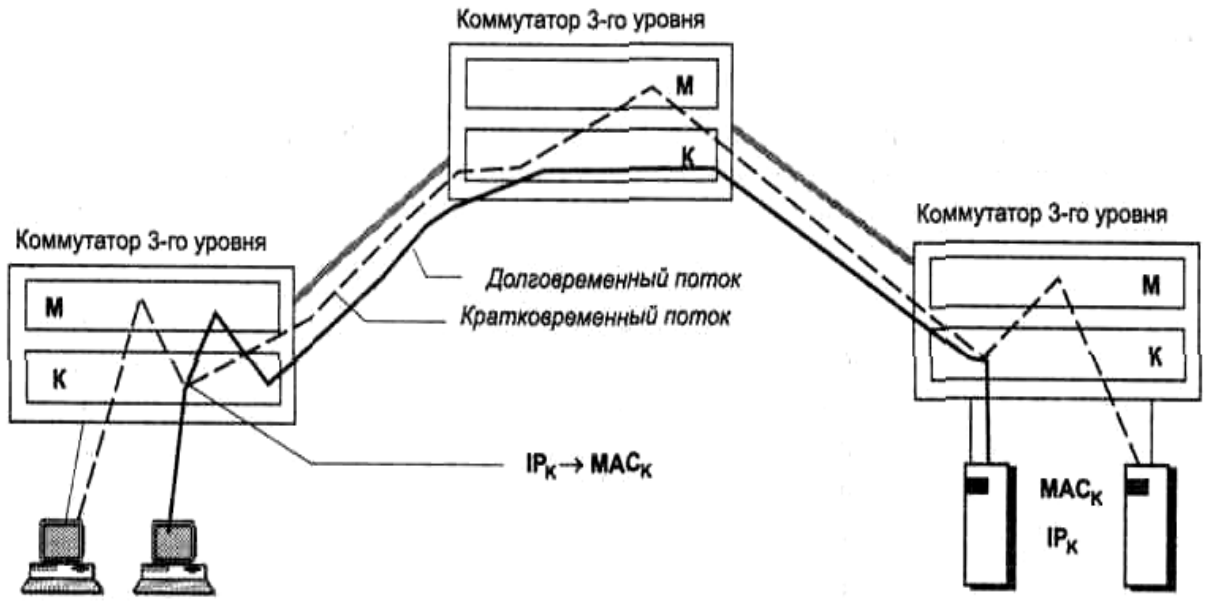


Рис. Маршрутизация потока пакетов

Если бы все коммутаторы 3-го уровня, изображенные на рис. , 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 коммутаторов. 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_K. Как только эта замена произведена, кадр с таким MAC-адресом перестает поступать на блоки маршрутизации второго и третьего коммутатора, 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 долговременности. Вторая проблема более серьезная. На основании какой информации первый коммутатор узнает MAC-адрес узла назначения? Этот узел непосредственно не подключен к сети первого коммутатора, 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 коммутаторы запрашивают этот MAC-адрес друг у друга, пока последний на пути коммутатор не выяснит его в своей сети, с помощью протокола ARP. Фирменные протоколы используют как распределенный подход, когда все коммутаторы равны в решении проблемы нахождения MAC-адреса, так и централизованный, когда в сети существует выделенный коммутатор, который помогает ее решить для всех.

Примерами коммутаторов 3-го уровня, работающими по схеме маршрутизации потоков, являются коммутаторы SmartSwitch компании Cabletron, а также коммутатор Catalyst 5000 компании Cisco, выполняющий свои функции совместно с маршрутизаторами Cisco 7500 по технологии Cisco NetFlow.

3.7. Шлюз (gateway), межсетевой экран (firewall), прокси-сервер, NAT.

Термин "шлюз" и термин "маршрутизатор" во многом схожи, но шлюз является более общим термином (всякий маршрутизатор является шлюзом). Шлюзом называется любое сетевое устройство, которое одновременно подключено к нескольким сетям при помощи нескольких сетевых интерфейсов, имеет в каждой сети свой адрес сетевого уровня и занимается продвижением пакетов между этими сетями. Например, шлюзом является компьютер одна сетевая карта которого подключена к сети 192.168.28.10.0 и имеет там IP-адрес 192.168.28.10.1, а другая сетевая карта подключена к сети 172.16.0.0 и имеет там IP-адрес 172.16.1.1. Шлюзом также будет являться и маршрутизатор. Даже обычный домашний компьютер, имеющий сетевую карту и модем можно рассматривать как шлюз, т.к. он имеет два интерфейса: один интерфейс – это интерфейс сетевой карты (локальной сети), IP-адрес которого может быть произвольным, а второй интерфейс – это интерфейс удаленного доступа (Internet), IP-адрес которого определяется провайдером, при подключении к нему по протоколу PPP.

Шлюз выполняет функции маршрутизации и продвижения пакетов между интерфейсами. Шлюзы также позволяют объединять разнородные (гетерогенные) сети, преобразуя, например, кадры Ethernet в кадры FDDI. Шлюз также является средством обеспечения безопасности подсети. Если сегмент сети соединен с остальной сетью через шлюз, то на шлюзе может быть установлен межсетевой экран (firewall, брандмауэр) – специальное программное обеспечение, которое контролирует как пакеты выходящие из данного сегмента, так и пакеты поступающие в данный сегмент. Межсетевой экран с фильтрацией пакетов уже был рассмотрен ранее (см. маршрутизаторы). Путем написания специальных правил, можно ограничить разрешенное взаимодействие между компьютерам сети и компьютерами сегмента. Правила имеют вид: "через шлюз допускается прохождение пакетов с IP-адресом отправителя 172.18.10.1 (порт 80) и IP-адресом получателя 192.168.1.1 (порт 21), в четверг с 15.00 до 19.00". Пакеты, не удовлетворяющие правилам фильтрации отбрасываются, а факт их наличия регистрируется в специальном журнале. Поскольку сервисы в сети связаны с определенными номерами портов, то закрыв входящие соединения на 23 порт, можно запретить извне управлять компьютерами сегмента по протоколу Telnet, а закрыв исходящие соединения на 80 порт можно запретить сотрудникам отдела (сегмента) просматривать Web-страницы.

Межсетевые экраны с фильтрацией пакетов просты, и в ряде случаев входят в состав самой операционной системы (например IPChains в ОС Linux). Однако межсетевые экраны с фильтрацией пакетов имеют и ряд недостатков:

- возможно задавать правила фильтрации по IP-адресам компьютеров, но не по имени пользователя.
- подсеть "видна" (маршрутизируется) извне.
- при выходе из строя межсетевого экрана подсеть становится незащищенной.

Для преодоления этих недостатков, в качестве межсетевого экрана используют прокси-сервера (proxy server). Прокси-сервер - это сервер посредник. Одно из назначений прокси-сервера - это ускорение работы сети, при подключении ее к Internet. Так например, кэширующий прокси-сервер Squid в ОС Linux сохраняет в достаточно большом кэше на диске Web-страницы, просмотренные разными пользователями, так что если какой-либо пользователь обратится к просмотренной кем-либо ранее странице, то эта страница не будет заново загружаться через Internet, а будет взята из кэша Squid. Прокси-сервер может использоваться и для сокрытия личности пользователя, путешествующего по Internet. Для этого пользователь сначала соединяется с анонимным прокси-сервером (например в Новой Зеландии), который получает пакеты от пользователя и перенаправляет их дальше от своего имени.

Установка прокси-сервера на шлюзе позволяет скрыть структуру подсети от внешней сети и реализовать гибкий межсетевой экран. При запрете продвижения IP-пакетов между интерфейсами шлюза, вся подсеть, с точки зрения внешней сети, представлена только одним IP-адресом – адресом прокси-сервера. Пользователь внешней сети, который хочет соединиться с компьютером внутри подсети, должен пройти следующую процедуру:

- Установить соединение с определенным портом прокси-сервера и указать имя компьютера внутри подсети с которым необходимо соединиться. Для каждого вида сервиса (http, ftp, smtp и т.д.) должна существовать своя программа-посредник, "прослушивающая" свой порт. Может существовать и универсальная программа - посредник, обслуживающая несколько сервисов. Если для какого-то сервиса программы-посредника нет (или она вышла из строя), то данный сервис будет не доступен. При установлении соединения возможно, хотя и не обязательно, проведение аутентификации (подтверждения личности) пользователя по его имени, паролю, IP-адресу. Возможна также регистрация факта и времени подключения в специальном журнале.
- Прокси-сервер создает соединение с компьютером внутри подсети, а затем обеспечивает обмен пакетами между внешней сетью и подсетью, подменяя адреса в проходящих через него пакетах. Возможно протоколирование соединения и фильтрация передаваемых данных (поскольку программа-посредник "понимает" протокол прикладного уровня своего сервиса).

Аналогичным образом происходит и соединение подсеть – внешняя сеть.

Другой технологией, позволяющей скрыть сеть предприятия, являеется NAT (Network Address Translation – трансляция сетевых адресов). NAT также позволяет компьютерам локальной сети работать с

Internet через один IP-адрес. Технология осуществляет подмену IP-адресов отправителя и получателя, в проходящих через шлюз пакетах.

Поясним принцип работы NAT на примере. Допустим, имеется локальная сеть из десяти компьютеров. Все компьютера в сети имеют "серые" адреса 192.168.1.1 – 192.168.1.20, которые изолированы от сети Internet (не передаются маршрутизаторами Internet) и их не надо согласовывать с InterNIC. На компьютере "А", с IP-адресом 192.168.1.1 имеется модем, сетевой интерфейс которого, при подключении к Internet по протоколу PPP, автоматически получает от провайдера IP-адрес w1.x1.y1.z1. (запись условная). Таким образом, компьютер "А" имеет два сетевых интерфейса с адресами 192.168.1.1 и w1.x1.y1.z1. и является шлюзом локальная сеть – Internet.

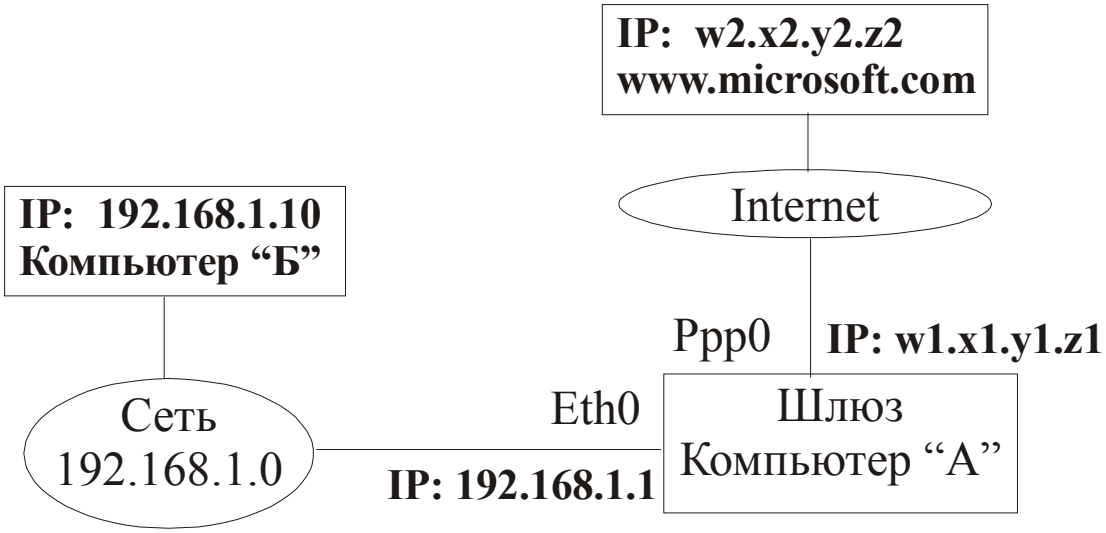


Рис. Технология NAT.

Пусть пользователь компьютера "Б" локальной сети (IP-адрес 192.168.1.10) обращается с помощью Web-браузера Internet Explorer к серверу www.microsoft.com, с IP-адресом w2.x2.y2.z2. (запись условная). Поскольку IP-адрес w2.x2.y2.z2. не относится к локальной сети, то на шлюз (компьютер "А") будет направлен IP-пакет со следующими данными:

- IP-адрес приемника: w2.x2.y2.z2 (www.microsoft.com)
- IP-адрес источника: 192.168.1.10 (компьютер "Б")
- Порт приемника: TCP-порт 80 (порт сервера Microsoft, протокол http)
- Порт источника: TCP-порт 1025 (порт компьютера "Б")

Этот IP-пакет перенаправляется протоколу NAT, который преобразовывает адреса исходящего пакета следующим образом.

- IP-адрес приемника: w2.x2.y2.z2 (www.microsoft.com)
- IP-адрес источника: w1.x1.y1.z1 (компьютер "А")
- Порт приемника: TCP-порт 80 (порт сервера Microsoft, протокол http)
- Порт источника: TCP-порт 5000 (порт компьютера "А")

При этом протокол NAT сохраняет в своей таблице преобразованных адресов запись:
"Адрес {192.168.1.10, порт 1025} заменен на адрес {w1.x1.y1.z1, порт 5000}".
Преобразованный IP-пакет отправляется по Интернету. Пакет, посылаемый в ответ на этот пакет, принимается протоколом NAT. Полученный пакет содержит следующие адресные данные.

- IP-адрес приемника: w1.x1.y1.z1 (компьютер "А")
- IP-адрес источника: w2.x2.y2.z2 (www.microsoft.com)
- Порт приемника: TCP-порт 5000 (порт компьютера "А")
- Порт источника: TCP-порт 80 (порт сервера Microsoft, протокол http)

Протокол NAT проверяет свою таблицу преобразованных адресов, после чего делает обратную замену:

- IP-адрес приемника: 192.168.1.10 (компьютер "Б")
- IP-адрес источника: w2.x2.y2.z2 (www.microsoft.com)
- Порт приемника: TCP-порт 1025 (порт компьютера "Б")
- Порт источника: TCP-порт 80 (порт сервера Microsoft, протокол http)

Преобразованный пакет направляется в локальную сеть.
Таким образом, протокол NAT подменяет адрес в IP-пакете, передавая его от своего имени, и пользуясь номером порта для того, чтобы "запомнить" какому компьютеру надо будет вернуть ответ на этот пакет. Шлюз не может одновременно создать с одного и того же своего порта два соединения с 80 портом сервера www.microsoft.com, т.к. для создания сокета необходимо, чтобы хотя бы один из параметров " IP-адрес отправителя, номер порта отправителя" – "IP-адрес получателя, номер порта получателя" у двух сетевых соединений не совпадали. Поэтому, если к серверу www.microsoft.com одновременно обратятся два

компьютера локальной сети, то пакеты одного из них будут отправлены с порта 5000 (как в примере), а пакеты другого – будут отправлены, например, с порта 5001. При получении ответа от сервера www.microsoft.com, пакеты, поступившие на порт 5000, будут переправлены первому компьютеру, а пакеты, поступившие на порт 5001 – второму компьютеру.

При использовании NAT возникает следующая проблема: пакеты, содержащие IP-адрес только в заголовке пакета, правильно преобразовываются протоколом NAT, однако пакеты, содержащие IP-адрес в поле данных, могут неправильно преобразовываться при помощи NAT. Например, протокол FTP хранит IP-адрес в заголовке FTP для команды FTP PORT. Заголовок FTP, как и любой протокол прикладного уровня, хранится в поле данных IP-пакета. Если NAT не сможет правильно преобразовать IP-адрес из заголовка FTP и откорректировать поле данных, то могут возникнуть неполадки связи. Для устранения этой проблемы существуют редакторы NAT, работающие с заголовками прикладных протоколов. Не для всех протоколов необходим редактор NAT, например для протокола HTTP он не нужен. В ОС Windows 2000 реализованы редакторы NAT для следующих протоколов: FTP, ICMP, PPTP, NetBIOS через TCP/IP, RPC, Direct Play, H.323, Регистрация ILS на основе LDAP. Однако для зашифрованного трафика использование редактора NAT не предусмотрено, что следует учитывать при использовании NAT.