6. Логическая структуризация сети с помощью мостов и коммутаторов

Под логической структуризацией сети понимается разбиение общей разделяемой среды на логические сегменты, которые представляют самостоятельные разделяемые среды с меньшим количеством узлов. Сеть, разделенная на логические сегменты, обладает более высокой производительностью и надежностью. Взаимодействие между логическими сегментами организуется с помощью мостов и коммутаторов.

6.1. Причины логической структуризации локальных сетей

Ограничения сети, построенной на общей разделяемой среде

Некоторое время после начала эксплуатации локальных сетей, при построении небольших сетей, состоящих из 10-30 узлов, использовались стандартные технологии на разделяемых средах передачи данных, что приводило к экономичным и эффективным решениям. Появление высокоскоростных технологий со скоростями обмена 100 и 1000 Мбит/с решило проблему качества транспортного обслуживания таких сетей.

Эффективность разделяемой среды для небольшой сети проявлялась в первую очередь в следующих свойствах:

• простой топологии сети, допускающей легкое наращивание числа узлов (в небольших пределах);

• отсутствии потерь кадров из-за переполнения буферов коммуникационных устройств, так как новый кадр не передается в сеть, пока не принят предыдущий — сама логика разделения среды регулирует поток кадров и приостанавливает станции, слишком часто генерирующие кадры, заставляя их ждать доступа;

• простоте протоколов, обеспечившей низкую стоимость сетевых адаптеров, повторителей и концентраторов.

Однако справедливым является и другое утверждение — крупные сети, насчитывающие сотни и тысячи узлов, не могут быть построены на основе одной разделяемой среды даже такой скоростной технологии, как Gigabit Ethernet. И не только потому, что практически все технологии ограничивают количество узлов в разделяемой среде: все виды семейства Ethernet — 1024 узлами, Token Ring — 260 узлами, а FDDI — 500 узлами. Даже сеть средних размеров, состоящая из 50-100 компьютеров и укладывающаяся в разрешенный максимум количества узлов, чаще всего будет плохо работать на одной разделяемой среде.

Основные недостатки сети на одной разделяемой среде начинают проявляться при превышении некоторого порога количества узлов, подключенных к разделяемой среде, и состоят в следующем. Даже та доля пропускной способности разделяемого сегмента, которая должна в среднем доставаться одному узлу (то есть, например, 10/N Мбит/с для сегмента Ethernet с N компьютерами), очень часто узлу не достается. Причина заключается в случайном характере метода доступа к среде, используемом в технологии Ethernet. В других технологиях, таких как Token Ring или FDDI, где метод доступа носит менее случайный характер, фактор доступа к среде все равно присутствует и оказывает свое негативное влияние на пропускную способность, достающуюся отдельному узлу.

На рис. 6.1 показана зависимость задержек доступа к среде передачи данных в сетях Ethernet, Token Ring и FDDI от коэффициента использования сети p, который также часто называют коэффициентом нагрузки сети. Напомним, что коэффициент использования сети равен отношению трафика, который должна передать сеть, к ее максимальной пропускной способности. Для сети Ethernet максимальная пропускная способность равна 10 Мбит/с, а трафик, который она должна передать, равен сумме интенсивностей трафика, генерируемого каждым узлом сети. Коэффициент использования обычно измеряют в относительных единицах или процентах.

Как видно из рисунка, всем технологиям присущ экспоненциальный рост величины задержек доступа при увеличении коэффициента использования сети, отличается только порог, при котором наступает резкий перелом в поведении сети, когда почти прямолинейная зависимость переходит в крутую экспоненту. Для всего семейства технологий Ethernet это 40-50 %, для технологии Token Ring — 60 %, а технологии FDDI- 70 %.

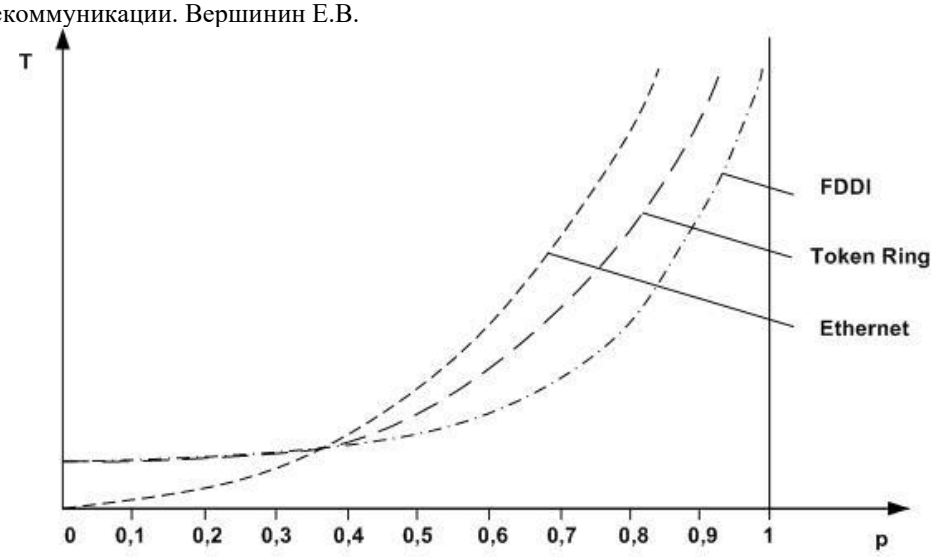


Рисунок 6.1. Задержки доступа к среде передачи данных

Количество узлов, при которых коэффициент использования сети начинает приближаться к опасной границе, зависит от типа функционирующих в узлах приложений. Если раньше для сетей Ethernet считалось, что 30 узлов — это вполне приемлемое число для одного разделяемого сегмента, то сегодня для мультимедийных приложений, перекачивающих большие объемы данных, эту цифру нужно уточнять с помощью натурных или имитационных экспериментов.

Влияние задержек и коллизий на полезную пропускную способность сети Ethernet хорошо отражает график, представленный на рис. 6.2.

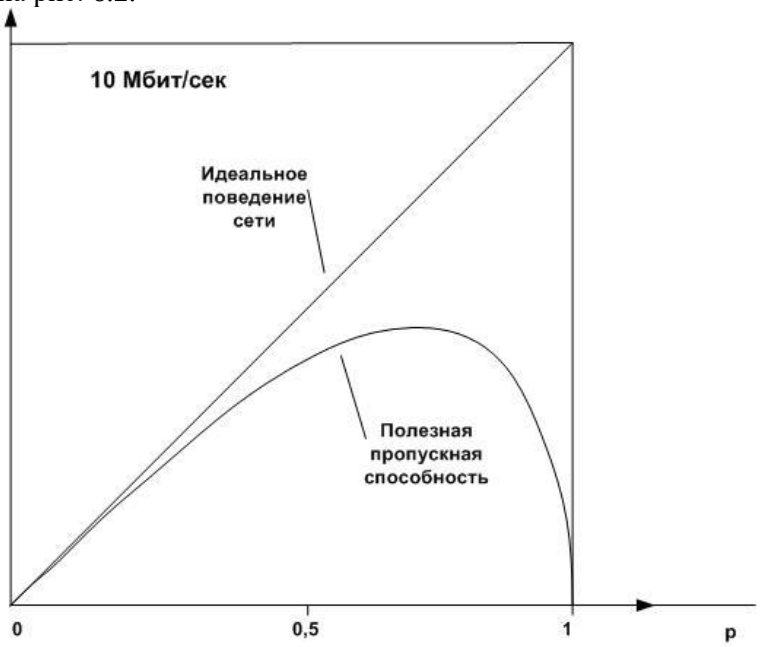


Рисунок 6.2. Зависимость полезной пропускной способности сети Ethernet от коэффициента использования

При загрузке сети до 50 % технология Ethernet на разделяемом сегменте хорошо справляется с передачей трафика, генерируемого конечными узлами. Однако при повышении интенсивности генерируемого узлами трафика сеть все больше времени начинает проводить неэффективно, повторно передавая кадры, которые вызвали коллизию. При возрастании интенсивности генерируемого трафика до такой величины, когда коэффициент использования сети приближается к 1, вероятность столкновения кадров настолько увеличивается, что практически любой кадр, который какая-либо станция пытается передать, сталкивается с другими кадрами, вызывая коллизию. Сеть перестает передавать полезную пользовательскую информацию и работает «на себя», обрабатывая коллизии.

Этот эффект хорошо известен на практике и исследован путем имитационного моделирования, поэтому сегменты Ethernet построенные на разделяемой среде не рекомендуется загружать так, чтобы среднее значение коэффициента использования превосходило 30 %.

Ограничения, связанные с возникающими коллизиями и большим временем ожидания доступа при значительной загрузке разделяемого сегмента, чаще всего оказываются более серьезными, чем ограничение на максимальное количество узлов, определенное в стандарте из соображений устойчивой

передачи электрических сигналов в кабелях.

В результате даже сеть средних размеров трудно построить на одном разделяемом сегменте так, чтобы она работала эффективно. Кроме того, при использовании разделяемой среды проектировщик сети сталкивается с жесткими ограничениями максимальной длины сети.

Преимущества логической структуризации сети

Ограничения, возникающие из-за использования общей разделяемой среды, можно преодолеть, разделив сеть на несколько сегментов и соединив их такими устройствами, как мосты, коммутаторы или маршрутизаторы (рис. 6.3).

Перечисленные устройства передают кадры с одного своего порта на другой, анализируя адрес назначения, помещенный в этих кадрах. (В отличие от концентраторов, которые повторяют кадры на всех своих портах, передавая их во все подсоединенные к ним сегменты, независимо от того, в каком из них находится станция назначения.) Мосты и коммутаторы выполняют операцию передачи кадров на основе плоских адресов канального уровня, то есть MAC-адресов, а маршрутизаторы — на основе номера сети. При этом единая разделяемая среда, созданная концентраторами (или в предельном случае — одним сегментом кабеля), делится на несколько частей, каждая из которых присоединена к порту моста, коммутатора или маршрутизатора.

Говорят, что при этом сеть делится на логические сегменты или сеть подвергается *логической структуризации.* Логический сегмент представляет собой единую разделяемую среду. Деление сети на логические сегменты приводит к тому, что нагрузка, приходящаяся па каждый из вновь образованных сегментов, почти всегда оказывается меньше, чем нагрузка, которую испытывала исходная сеть. Следовательно, уменьшаются вредные эффекты от разделения среды: снижается время ожидания доступа, а в сетях Ethernet — и интенсивность коллизий.

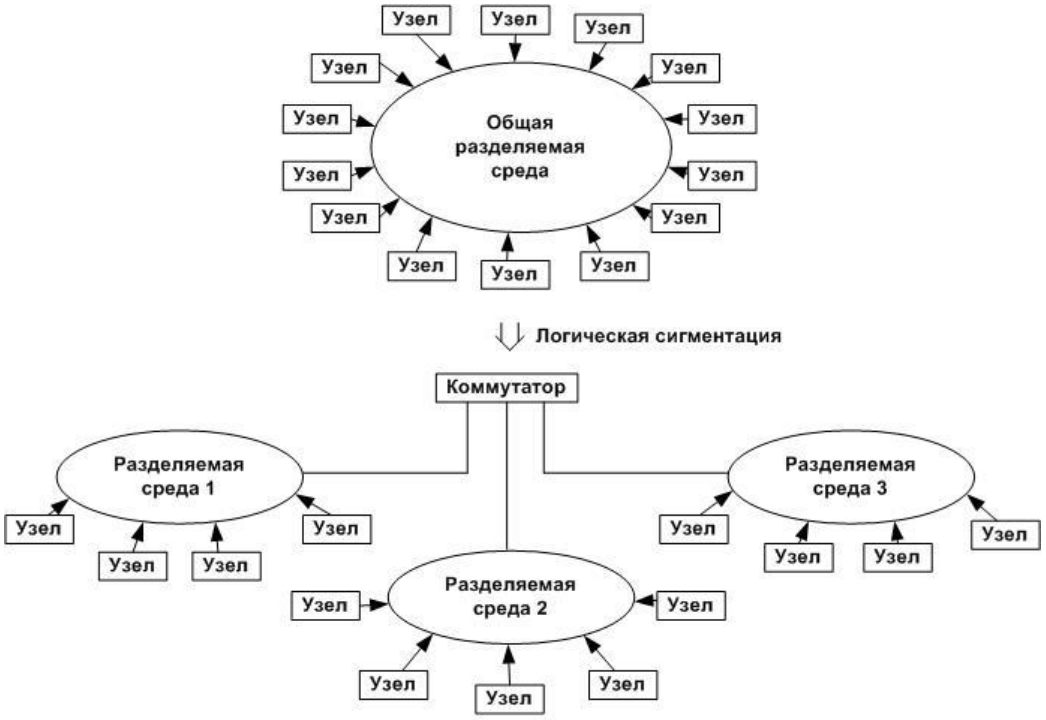


Рисунок 6.3. Логическая структуризация сети

Большинство крупных сетей разрабатывается на основе структуры с общей магистралью, к которой через мосты и маршрутизаторы присоединяются подсети. Эти подсети обслуживают различные отделы. Подсети могут делиться и далее на сегменты, предназначенные для обслуживания рабочих групп.

Сегментация увеличивает гибкость сети. При построении сети как совокупности подсетей каждая подсеть может быть адаптирована к специфическим потребностям рабочей группы или отдела. Например, в одной подсети может использоваться технология Ethernet, а в другой Token Ring, в соответствии с традициями того или иного отдела. Вместе с тем, у пользователей обеих подсетей есть возможность обмениваться данными через межсетевые устройства, такие как мосты, коммутаторы, маршрутизаторы. Процесс разбиения сети на логические сегменты можно рассматривать и в обратном направлении, как процесс создания большой сети из модулей — уже имеющихся подсетей.

Подсети повышают безопасность данных. При подключении пользователей к различным физическим сегментам сети можно запретить доступ определенных пользователей к ресурсам других сегментов. Устанавливая различные логические фильтры на мостах, коммутаторах и маршрутизаторах,

можно контролировать доступ к ресурсам, чего не позволяют сделать повторители.

Подсети упрощают управление сетью. Побочным эффектом уменьшения трафика и повышения безопасности данных является упрощение управления сетью. Проблемы очень часто локализуются внутри сегмента. Как и в случае структурированной кабельной системы, проблемы одной подсети не оказывают влияния на другие подсети. Подсети образуют логические домены управления сетью.

Сети должны проектироваться на двух уровнях: физическом и логическом. Логическое проектирование определяет места расположения ресурсов, приложений и способы группировки этих ресурсов в логические сегменты.

Структуризация с помощью мостов и коммутаторов

В данной лекции рассмотрим устройства логической структуризации сетей, работающие на канальном уровне стека протоколов, а именно — мосты и коммутаторы. Структуризация сети возможна также на основе маршрутизаторов, которые для выполнения этой задачи привлекают протоколы сетевого уровня. Каждый способ структуризации — с помощью канального протокола и с помощью сетевого протокола — имеет свои преимущества и недостатки. В современных сетях часто используют комбинированный способ логической структуризации — отдельные узлы сети объединяются устройствами канального уровня в подсети, которые, в свою очередь, соединяются маршрутизаторами.

Итак, сеть можно разделить на логические сегменты с помощью устройств двух типов — мостов (bridge) и/или коммутаторов (switch, switching hub). Мост и коммутатор — это функциональные близнецы. Оба эти устройства продвигают кадры на основании одних и тех же алгоритмов. Мосты и коммутаторы используют два типа алгоритмов: алгоритм *прозрачного моста (transparent bridge)*, описанного в стандарте IEEE 802.1D, либо алгоритм *моста с маршрутизацией от источника (source routing bridge)* компании IBM для сетей Token Ring. Эти стандарты были разработаны задолго до появления первого коммутатора, поэтому в них используется термин «мост». Когда же на свет появилась первая промышленная модель коммутатора для технологии Ethernet, то она выполняла тот же алгоритм продвижения кадров IEEE 802.1D, который был с десяток лет отработан мостами локальных и глобальных сетей. Точно так же поступают и все современные коммутаторы. Коммутаторы, которые продвигают кадры протокола Token Ring, работают по алгоритму Source Routing, характерному для мостов IBM.

Основное отличие коммутатора от моста заключается в том, что мост обрабатывает кадры последовательно, а коммутатор — параллельно. Это обстоятельство связано с тем, что мосты появились в те времена, когда сеть делили на небольшое количество сегментов, а межсегментный трафик был небольшим. Сеть чаще всего делили на два сегмента, поэтому и термин был выбран соответствующий — мост. Для обработки потока данных со средней интенсивностью 1 Мбит/с мосту вполне хватало производительности одного процессорного блока.

При изменении ситуации в конце 80-х — начале 90-х годов — появлении быстрых протоколов, производительных персональных компьютеров, мультимедийной информации, разделении сети на большое количество сегментов — классические мосты перестали справляться с работой. Обслуживание потоков кадров между теперь уже несколькими портами с помощью одного процессорного блока требовало значительного повышения быстродействия процессора, а это довольно дорогостоящее решение.

Более эффективным оказалось решение, которое и «породило» коммутаторы: для обслуживания потока, поступающего на каждый порт, в устройство ставился отдельный специализированный процессор, который реализовывал алгоритм моста. По сути, коммутатор — это мультипроцессорный мост, способный параллельно продвигать кадры сразу между всеми парами своих портов. Но если при добавлении процессорных блоков компьютер не перестали называть компьютером, а добавили только прилагательное «мультипроцессорный», то с мультипроцессорными мостами произошла метаморфоза — они превратились в коммутаторы. Этому способствовал способ связи между отдельными процессорами коммутатора — они связывались коммутационной матрицей, похожей на матрицы мультипроцессорных компьютеров, связывающие процессоры с блоками памяти.

Постепенно коммутаторы вытеснили из локальных сетей классические однопроцессорные мосты. Основная причина этого — очень высокая производительность, с которой коммутаторы передают кадры между сегментами сети. Если мосты могли даже замедлять работу сети, когда их производительность оказывалась меньше интенсивности межсегментного потока кадров, то коммутаторы всегда выпускаются с процессорами портов, которые могут передавать кадры с той максимальной скоростью, на которую рассчитан протокол. Добавление к этому параллельной передачи кадров между портами сделало производительность коммутаторов на несколько порядков выше, чем мостов. Это и предопределило судьбу мостов и коммутаторов.

Процесс вытеснения мостов начал протекать достаточно быстро с 1994 года, и сегодня локальные мосты не производятся сетевой индустрией. За время своего существования уже без конкурентов-мостов коммутаторы вобрали в себя многие дополнительные функции, которые появлялись в результате естественного развития сетевых технологий. К этим функциям относятся, например, поддержка виртуальных сетей (VLAN), приоритезация трафика, использование магистрального порта по умолчанию и т. п.

Прозрачные мосты умеют, кроме передачи кадров в рамках одной технологии, транслировать протоколы локальных сетей, например Ethernet в Token Ring, FDDI в Ethernet и т. п. Это свойство прозрачных мостов описано в стандарте IEEE 802.1H.

В дальнейшем будем называть устройство, которое продвигает кадры по алгоритму моста и работает в локальной сети, современным термином «коммутатор». При описании же самих алгоритмов 802.1D и Source Routing будем по традиции называть устройство мостом, как собственно оно в этих стандартах и называется.

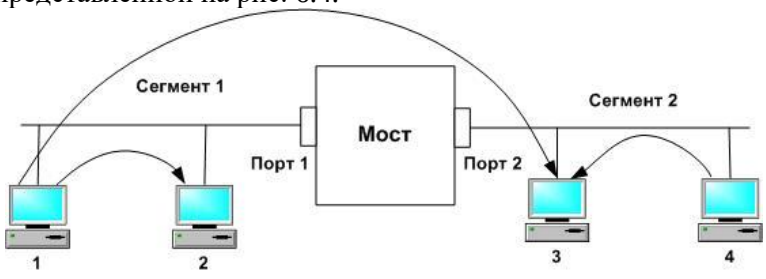
6.2. Принципы работы мостов

Алгоритм работы прозрачного моста

Прозрачные мосты незаметны для сетевых адаптеров конечных узлов, так как они самостоятельно строят специальную адресную таблицу, на основании которой можно решить, нужно передавать пришедший кадр в какой-либо другой сегмент или нет. Сетевые адаптеры при использовании прозрачных мостов работают точно так же, как и в случае их отсутствия, то есть не предпринимают никаких дополнительных действий, чтобы кадр прошел через мост. Алгоритм прозрачного моста не зависит от технологии локальной сети, в которой устанавливается мост, поэтому прозрачные мосты Ethernet работают точно так же, как прозрачные мосты FDDI.

Прозрачный мост строит свою адресную таблицу на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом мост учитывает адреса источников кадров данных, поступающих на порты моста. По адресу источника кадра мост делает вывод о принадлежности этого узла тому или иному сегменту сети.

Рассмотрим процесс автоматического создания адресной таблицы моста и ее использования на примере простой сети, представленной на рис. 6.4.



MAC-адрес	Порт
1	1
2	1
3	2
4	2

Рисунок 6.4. Принцип работы прозрачного моста

Мост соединяет два логических сегмента. Сегмент 1 составляют компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту 1 моста, а сегмент 2 — компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту 2 моста.

Каждый порт моста работает как конечный узел своего сегмента за одним исключением — порт моста не имеет собственного MAC-адреса. Порт моста работает в так называемом *неразборчивом (promiscuous)* режиме захвата пакетов, когда все поступающие на порт кадры запоминаются в буферной памяти. С помощью такого режима мост следит за всем трафиком, передаваемым в присоединенных к нему сегментах, и использует проходящие через него кадры для изучения состава сети. Так как в буфер записываются все кадры, то адрес порта мосту не нужен.

В исходном состоянии мост ничего не знает о том, компьютеры с какими MAC-адресами подключены к каждому из его портов. Поэтому в этом случае мост просто передает любой захваченный и буферизованный кадр на все свои порты за исключением того, от которого этот кадр получен. В

нашем примере у моста только два порта, поэтому он передает кадры с порта 1 на порт 2, и наоборот. Отличие работы моста в этом режиме от повторителя в том, что он передает кадр не побитно, а с буферизацией. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды. Когда мост собирается передать кадр с сегмента на сегмент, например с сегмента 1 на сегмент 2, он заново пытается получить доступ к сегменту 2 как конечный узел по правилам алгоритма доступа, в данном примере — по правилам алгоритма CSMA/CD.

Одновременно с передачей кадра на все порты мост изучает адрес источника кадра и делает новую запись о его принадлежности в своей адресной таблице, которую также называют таблицей фильтрации или маршрутизации. Например, получив на свой порт 1 кадр от компьютера 1, мост делает первую запись в своей адресной таблице: MAC-адрес 1 — порт 1. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро мост построит полную адресную таблицу сети, состоящую из 4 записей — по одной записи на узел.

После того как мост прошел этап обучения, он может работать более рационально. При получении кадра, направленного, например, от компьютера 1 компьютеру 3, он просматривает адресную таблицу на предмет совпадения ее адресов с адресом назначения 3. Поскольку такая запись есть, то мост выполняет второй этап анализа таблицы — проверяет, находятся ли компьютеры с адресами источника (в нашем случае — это адрес 1) и адресом назначения (адрес 3) в одном сегменте. Так как в нашем примере они находятся в разных сегментах, то мост выполняет операцию *продвижения (forwarding)* кадра — передает кадр на другой порт, предварительно получив доступ к другому сегменту.

Если бы оказалось, что компьютеры принадлежат одному сегменту, то кадр просто был бы удален из буфера и работа с ним на этом бы закончилась. Такая операция называется *фильтрацией (filtering).*

Если же адрес назначения неизвестен, то мост передает кадр на все свои порты, кроме порта — источника кадра, как и на начальной стадии процесса обучения.

На самом деле мы несколько упростили алгоритм работы моста. Его процесс обучения никогда не заканчивается. Мост постоянно следит за адресами источника буферизуемых кадров, чтобы быть в состоянии автоматически приспосабливаться к изменениям, происходящим в сети, — перемещениям компьютеров из одного сегмента сети в другой, появлению новых компьютеров. С другой стороны, мост не ждет, когда адресная таблица заполнится полностью (да это и невозможно, поскольку заранее не известно, сколько компьютеров и адресов будут находиться в сегментах моста). Как только в таблице появляется первый адрес, мост пытается его использовать, проверяя совпадение с ним адресов назначения всех поступающих кадров.

Записи адресной таблицы могут быть динамическими, создаваемыми в процессе самообучения моста, и статическими, создаваемыми вручную администратором сети. Динамические записи имеют срок жизни — при создании или обновлении записи в адресной таблице с ней связывается отметка времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время мост не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность автоматически реагировать на перемещения компьютера из сегмента в сегмент — при его отключении от старого сегмента запись о его принадлежности к нему со временем вычеркивается из адресной таблицы. После включения этого компьютера в работу в другом сегменте его кадры начнут попадать в буфер моста через другой порт, и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Статические записи не имеют срока жизни, что дает администратору возможность подправлять работу моста, если это необходимо.

Кадры с широковещательными MAC-адресами передаются мостом на все его порты, как и кадры с неизвестным адресом назначения. Такой режим распространения кадров называется *затоплением сети (flood).* Наличие мостов в сети не препятствует распространению широковещательных кадров по всем сегментам сети, сохраняя ее прозрачность. Однако это является достоинством только в том случае, когда широковещательный адрес выработан корректно работающим узлом. Однако часто случается так, что в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сам сетевой адаптер начинают работать некорректно и постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом в течение длительного промежутка времени. Мост в этом случае передает эти кадры во все сегменты, затапливая сеть ошибочным трафиком. Такая ситуация называется *широковещательным штормом (broadcast storm).*

К сожалению, мосты не защищают сети от широковещательного шторма, во всяком случае, по умолчанию, как это делают маршрутизаторы. Максимум, что может сделать администратор с помощью моста для борьбы с широковещательным штормом — установить для каждого узла предельно допустимую интенсивность генерации кадров с широковещательным адресом. Но при этом нужно точно знать, какая интенсивность является нормальной, а какая — ошибочной. При смене протоколов ситуация в сети может измениться, и то, что вчера считалось ошибочным, сегодня может оказаться

нормой. Таким образом, мосты располагают весьма грубыми средствами борьбы с широковещательным штормом.

На рис. 6.5 показана типичная структура моста. Функции доступа к среде при приеме и передаче кадров выполняют микросхемы MAC, которые идентичны микросхемам сетевого адаптера.

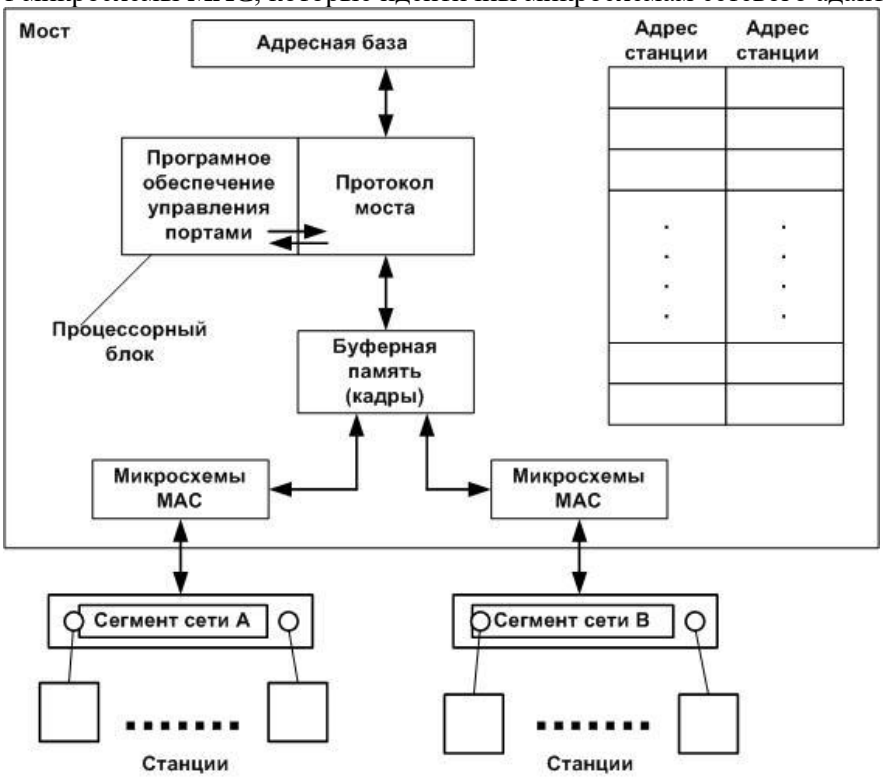


Рисунок 6.5. Структурная схема моста

Мосты с маршрутизацией от источника

Мосты с маршрутизацией от источника применяются для соединения колец Token Ring и FDDI, хотя для этих же целей могут использоваться и прозрачные мосты. Маршрутизация от источника (Source Routing, SR) основана на том, что станция-отправитель помещает в посылаемый в другое кольцо кадр всю адресную информацию о промежуточных мостах и кольцах, которые должен пройти кадр перед тем, как попасть в кольцо, к которому подключена станция-получатель. Хотя в название этого способа входит термин «маршрутизация», настоящей маршрутизации в строгом понимании этого термина здесь нет, так как мосты и станции по-прежнему используют для передачи кадров данных только информацию MAC-уровня, а заголовки сетевого уровня для мостов данного типа по-прежнему остаются неразличимой частью поля данных кадра.

Рассмотрим принципы работы мостов Source Routing (в дальнейшем, SR-мосты) на примере сети, изображенной на рис. 6.6. Сеть состоит из трех колец, соединенных тремя мостами. Для задания маршрута кольца и мосты имеют идентификаторы. SR-мосты не строят адресную таблицу, а при продвижении кадров пользуются информацией, имеющейся в соответствующих полях кадра данных.

Рисунок 6.6. Мосты работающие по алгоритму Source Routing

При получении каждого кадра SR-мосту нужно только просмотреть поле маршрутной информации (поле Routing Information Field, RIF, в кадре Token Ring или FDDI) на предмет наличия в нем своего идентификатора. И если он там присутствует и сопровождается идентификатором кольца, которое подключено к данному мосту, то в этом случае мост копирует поступивший кадр в указанное кольцо. В противном случае кадр в другое кольцо не копируется. В любом случае исходная копия кадра возвращается по исходному кольцу станции-отправителю, и если он был передан в другое кольцо, то бит А (адрес распознан) и бит С (кадр скопирован) поля статуса кадра устанавливаются в 1, чтобы сообщить станции-отправителю, что кадр был получен станцией назначения (в данном случае передан мостом в другое кольцо).

Так как маршрутная информация в кадре нужна не всегда, а только для передачи кадра между станциями, подключенными к разным кольцам, то наличие в кадре поля RIF обозначается установкой в 1 бит индивидуального/группового(I/G) адреса отправителя (при этом данный бит используется не по назначению, так как адрес источника всегда индивидуальный).

Поле RIF имеет управляющее подполе, состоящее из трех частей.

• *Тип кадра* определяет тип поля RIF. Существуют различные типы полей RIF, использующиеся для нахождения маршрута и для отправки кадра по известному маршруту.

• *Поле максимальной длины кадра* используется мостом для связи колец, в которых установлено различное значение MTU. С помощью этого поля мост уведомляет станцию о максимально возможной длине кадра (то есть минимальном значении MTU на протяжении всего составного маршрута).

• *Длина поля RIF* необходима, так как заранее неизвестно количество описателей маршрута, задающих идентификаторы пересекаемых колец и мостов.

Для работы алгоритма маршрутизации от источника используются два дополнительных типа кадра — одномаршрутный широковещательный кадр-исследователь SRBF (single-route broadcast frame) и многомаршрутный широковещательный кадр-исследователь ARBF (all-route broadcast frame).

Все SR-мосты должны быть сконфигурированы администратором вручную, чтобы передавать кадры ARBF на все порты, кроме порта-источника кадра, а для кадров SRBF некоторые порты мостов нужно заблокировать, чтобы в сети не было петель. В примере сети на рис. 6.6 для исключения петли администратор заблокировал оба порта моста 3 для передачи кадров SRBF,

Кадр первого типа отправляется станцией, когда она, во-первых, определяет, что станция назначения находится в другом кольце, а во-вторых, ей неизвестно, через какие мосты и кольца пролегает путь к этой станции назначения, то есть неизвестен маршрут до этой станции. Первое обстоятельство выясняется, если кадр, отправленный по кольцу, возвращается в станцию-источник с неустановленными признаками распознавания адреса и копирования. Значит, ни одна из станций исходного кольца не является станцией назначения, и кадр надо передавать по некоторому составному маршруту. Отсутствие маршрута к станции назначения в таблице моста является вторым обстоятельством, которое и вызывает отправку одномаршрутного кадра-исследователя SRBF.

В кадре SRBF станция задает длину поля RIF, равную нулю. Как и прозрачные мосты, SR-мосты работают в режиме «неразборчивого» захвата, буферизуя и анализируя все кадры. При получении кадра SRBF SR-мост передает его в исходном виде на все незаблокированные для этого типа кадров порты. Необходимость в конфигурировании топологии без петель для кадров-исследователей SRBF вызвана тем, что таким способом предотвращается возможность бесконечного зацикливания этих кадров.

В конце концов, кадр-исследователь SRBF, распространяясь по всем кольцам сети, доходит до станции назначения. В ответ станция назначения отправляет многомаршрутный широковещательный кадр-исследователь ARBF станции-отправителю, В отличие от кадра SRBF этот кадр передается мостами через все порты. При приеме такого кадра каждый промежуточный мост добавляет в поле маршрутной информации RIF новый описатель маршрута (свой идентификатор и идентификатор сегмента, от которого получен кадр), наращивает длину поля маршрутной информации и широковещательно его распространяет.

Для предотвращения зацикливания кадров ARBF мосты обрабатывают их следующим образом. Перед передачей кадра на какой-либо сегмент мост проверяет, нет ли идентификатора этого сегмента в списке маршрутов кадра. Если такой сегмент уже был пройден кадром, то кадр в данный сегмент не направляется.

Станция-источник получает в общем случае несколько кадров-ответов, прошедших по всем возможным маршрутам составной сети, и выбирает наилучший маршрут (обычно по количеству пересечений промежуточных мостов). Именно для получения информации о всех возможных маршрутах кадр ARBF передается по всем возможным направлениям.

Затем маршрутная информация помещается в таблицу маршрутизации станции и используется для отправки кадров данных станции назначения по наилучшему маршруту за счет помещения

последовательности номеров сетей и мостов в заголовке каждого такого кадра.

Мосты с маршрутизацией от источника имеют по сравнению с прозрачными мостам как преимущества, так и недостатки, отраженные в табл. 6.1.

Наличие двух возможных алгоритмов работы мостов — от источника и в прозрачном режиме — создает трудности для построения сложных сетей Token Ring. Мосты, работающие от источника, не могут поддерживать сегменты, рассчитанные на работу в прозрачном режиме, и наоборот.

До некоторого времени эта проблема решалась двумя способами. Один способ заключался в использовании во всех сегментах либо только маршрутизации от источника, либо только прозрачных мостов. Другим способом была установка маршрутизаторов. Позже появилось третье решение. Оно основано на стандарте, который позволяет объединить обе технологии работы моста в одном устройстве. Этот стандарт, называемый SRT (Source Route Transparent), позволяет мосту работать в любом режиме. Мост просматривает специальные флаги в заголовке кадров Token Ring и автоматически определяет, какой из алгоритмов нужно применить.

Таблица 6.1. Преимущества и недостатки мостов с маршрутизацией источника

Преимущества	Недостатки
Более рациональные маршруты	Более дорогие сетевые адаптеры, принимающие участие в маршрутизации
Проще и дешевле - не нужно строить таблицы фильтрации	Сеть непрозрачна - кольца имеют номера
Более высокая скорость - не нужно просматривать таблицы фильтрации	Увеличивается трафик за счет шировещательных каналов

Ограничения топологии сети, построенной на мостах

Слабая защита от широковещательного шторма — одно из главных ограничений моста, но не единственное. Другим серьезным ограничением является невозможность поддержки петлеобразных конфигураций сети. Рассмотрим это ограничение на примере сети, изображенной на рис. 6.7.

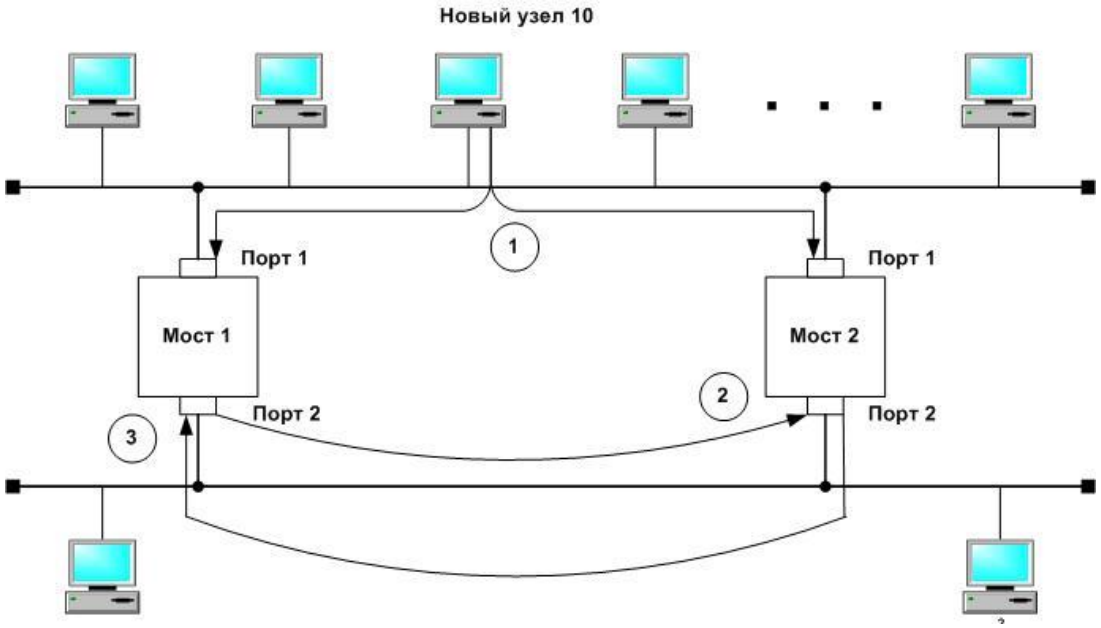


Рисунок 6.7. Влияние замкнутых маршрутов на работу мостов

Два сегмента параллельно соединены двумя мостами, так что образовалась активная петля. Пусть новая станция с адресом 10 впервые начинает работу в данной сети. Обычно начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых станция заявляет о своем существовании и одновременно ищет серверы сети.

На этапе 1 станция 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	Порт
10	1

Так как адрес назначения широковещательный, то каждый мост должен передать кадр на сегмент 2. Эта передача происходит поочередно, в соответствии с методом случайного доступа технологии Ethernet. Пусть первым доступ к сегменту 2 получил мост 1 (этап 2 на рис. 6.7). При появлении кадра на сегменте 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.

Теперь адресная таблица моста 2 будет иметь уже другую запись о станции с адресом 10:

MAC-адрес	Порт
10	1
10	2

Аналогично поступает мост 1, когда мост 2 передает свою копию кадра на сегмент 2.

Результаты наличия петли перечислены ниже.

• «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. Тогда кадры от каждой станции будут поступать в мост всегда с одного и того же порта, и мост сможет правильно решать задачу выбора рационального маршрута в сети.

Ограничение топологии структурированной сети древовидной структурой вытекает из самого принципа построения адресной таблицы мостом, а поэтому точно так же это ограничение действует и на коммутаторы.

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 коммутаторов. 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 коммутаторов.

6.3. Коммутаторы локальных сетей

Технология коммутации сегментов Ethernet была предложена фирмой Kalpana в 1990 году в ответ на растущие потребности в повышении пропускной способности связей высокопроизводительных серверов с сегментами рабочих станций.

Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana, представлена на рис. 6.8.

Каждый из 8 портов l0Base-T обслуживался одним процессором пакетов Ethernet — EPP (Ethernet Packet Processor). Кроме того, коммутатор имел системный модуль, который координировал работу всех процессоров EPP. Системный модуль вел общую адресную таблицу коммутатора и обеспечивал управление коммутатором по протоколу SNMP. Для передачи кадров между портами использовалась коммутационная матрица, подобная тем, которые работают в телефонных коммутаторах или мультипроцессорных компьютерах, соединяя несколько процессоров с несколькими модулями памяти.

Коммутационная матрица работает по принципу коммутации каналов. Для 8 портов матрица может обеспечить 4 одновременных внутренних канала при полудуплексном режиме работы портов и 8 — при полнодуплексном, когда передатчик и приемник каждого порта работают независимо друг от друга.

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, 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, которую тот буферизует в своем кэше для последующего использования.

Рисунок 6.8. Структура коммутатора EtherSwitch

После нахождения адреса назначения процессор EPP знает, что нужно дальше делать с поступающим кадром (во время просмотра адресной таблицы процессор продолжал буферизацию поступающих в порт байтов кадра). Если кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра.

Если же кадр нужно передать на другой порт, то процессор обращается к коммутационной матрице и пытается установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу назначения. Коммутационная матрица может это сделать только в том случае, когда порт адреса назначения в этот момент свободен, то есть не соединен с другим портом.

Если же порт занят, то, как и в любом устройстве с коммутацией каналов, матрица в соединении отказывает. В этом случае кадр полностью буферизуется процессором входного порта, после чего процессор ожидает освобождения выходного порта и образования коммутационной матрицей нужного пути.

После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet (в случае полудуплексного режима работы порта по алгоритму CSMA/CD), байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байт принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра (рис. 6.9).

При свободном в момент приема кадра состоянии выходного порта задержка между приемом первого байта кадра коммутатором и появлением этого же байта на выходе порта адреса назначения составляла у коммутатора компании Kalpana всего 40 мкс, что было гораздо меньше задержки кадра при его передаче мостом.

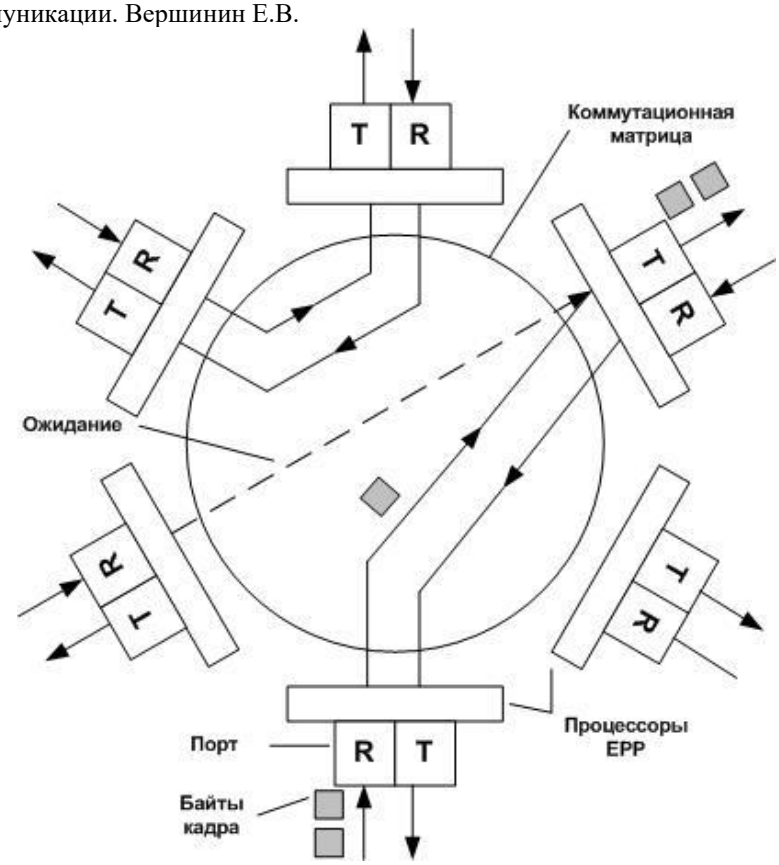


Рисунок 6.9. Передача кадра через коммутационную матрицу

Описанный способ передачи кадра без его полной буферизации получил название коммутации «на лету» («on-the-fly») или «напролет» («cut-through»). Этот способ представляет, по сути, конвейерную обработку кадра, когда частично совмещаются во времени несколько этапов его передачи (рис. 6.10).

1. Прием первых байт кадра процессором входного порта, включая прием байт адреса назначения.
2. Поиск адреса назначения в адресной таблице коммутатора (в кэше процессора или в общей таблице системного модуля).
3. Коммутация матрицы.
4. Прием остальных байт кадра процессором входного порта.
5. Прием байт кадра (включая первые) процессором выходного порта через коммутационную матрицу.
6. Получение доступа к среде процессором выходного порта.
7. Передача байт кадра процессором выходного порта в сеть.

Этапы 2 и 3 совместить во времени нельзя, так как без знания номера выходного порта операция коммутации матрицы не имеет смысла.

По сравнению с режимом полной буферизации кадра, также приведенном на рис. 6.10, экономия от конвейеризации получается ощутимой.

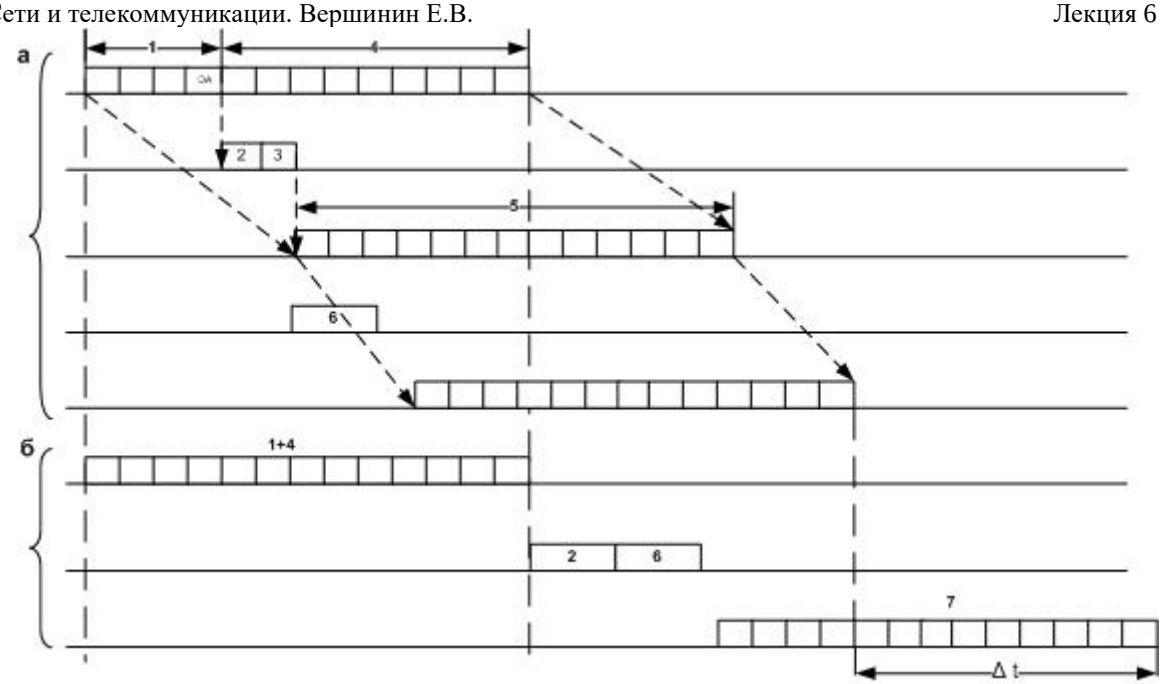


Рисунок 6.10. Затраты времени при работе коммутатора в режимах:
а – на лету, б – с полной буферизацией

Однако главной причиной повышения производительности сети при использовании коммутатора является *параллельная* обработка нескольких кадров.

Этот эффект иллюстрирует рис. 6.11. На рисунке изображена идеальная в отношении повышения производительности ситуация, когда четыре порта из восьми передают данные с максимальной для протокола Ethernet скоростью 10 Мб/с, причем они передают эти данные на остальные четыре порта коммутатора не конфликтуя — потоки данных между узлами сети распределились так, что для каждого принимающего кадры порта есть свой выходной порт. Если коммутатор успевает обрабатывать входной трафик даже при максимальной интенсивности поступления кадров на входные порты, то общая производительность коммутатора в приведенном примере составит 4x10 = 40 Мбит/с, а при обобщении примера для N портов — (N/2)x10 Мбит/с. Говорят, что коммутатор предоставляет каждой станции или сегменту, подключенным к его портам, выделенную пропускную способность протокола.

Естественно, что в сети не всегда складывается такая ситуация, которая изображена на рис. 4.26. Если двум станциям, например станциям, подключенным к портам *3* и *4,* одновременно нужно записывать данные на один и тот же сервер, подключенный к порту *8,* то коммутатор не сможет выделить каждой станции поток данных по 10 Мбит/с, так как порт *8* не может передавать данные со скоростью 20 Мбит/с. Кадры станций будут ожидать во внутренних очередях входных портов *3* и *4,* когда освободится порт *8* для передачи очередного кадра. Очевидно, хорошим решением для такого распределения потоков данных было бы подключение сервера к более высокоскоростному порту, например Fast Ethernet.

Так как главное достоинство коммутатора, благодаря которому он завоевал очень хорошие позиции в локальных сетях, это его высокая производительность, то разработчики коммутаторов стараются выпускать так называемые *неблокирующие (non-blocking)* модели коммутаторов.

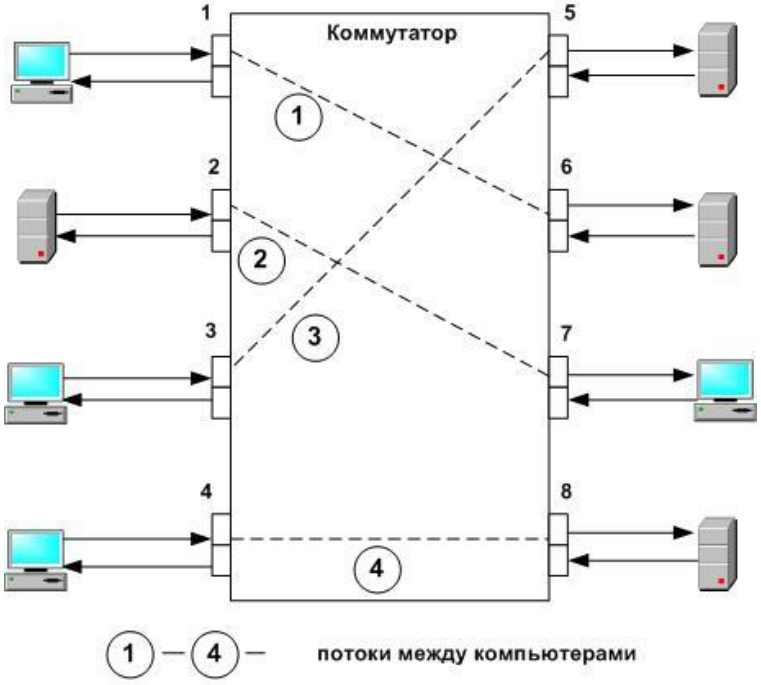


Рисунок 6.11. Параллельная передача кадров коммутатором

Неблокирующий коммутатор — это такой коммутатор, который может передавать кадры через свои порты с той же скоростью, с которой они на них поступают. Естественно, что даже неблокирующий коммутатор не может разрешить в течение долгого промежутка времени ситуации, подобные описанной выше, когда блокировка кадров происходит из-за ограниченной скорости выходного порта.

Обычно имеют в виду устойчивый неблокирующий режим работы коммутатора, когда коммутатор передает кадры со скоростью их поступления в течение произвольного промежутка времени. Для обеспечения такого режима нужно, естественно, такое распределение потоков кадров по выходным портам, чтобы они справлялись с нагрузкой и коммутатор мог всегда в среднем передать на выходы столько кадров, сколько их поступило на входы. Если же входной поток кадров (просуммированный по всем портам) в среднем будет превышать выходной поток кадров (также просуммированный по всем портам), то кадры будут накапливаться в буферной памяти коммутатора, а при превышении ее объема — просто отбрасываться. Для обеспечения неблокирующего режима коммутатора необходимо выполнение достаточно простого условия:

$Ck = (\sum Cpi)/2,$

где Ck — производительность коммутатора, Cpi — максимальная производительность протокола, поддерживаемого i-м портом коммутатора. Суммарная производительность портов учитывает каждый проходящий кадр дважды — как входящий кадр и как выходящий, а так как в устойчивом режиме входной трафик равен выходному, то минимально достаточная производительность коммутатора для поддержки неблокирующего режима равна половине суммарной производительности портов. Если порт работает в полудуплексном режиме, например Ethernet 10 Мбит/с, то производительность порта Cpi равна 10 Мбит/с, а если в полнодуплексном, то его Cpi будет составлять 20 Мбит/с.

Иногда говорят, что коммутатор поддерживает мгновенный неблокирующий режим. Это означает, что он может принимать и обрабатывать кадры от всех своих портов на максимальной скорости протоколов, независимо от того, обеспечиваются ли условия устойчивого равновесия между входным и выходным трафиком. Правда, обработка некоторых кадров при этом может быть неполной — при занятости выходного порта кадр помещается в буфер коммутатора. Для поддержки неблокирующего мгновенного режима коммутатор должен обладать большей собственной производительностью, а именно, она должна быть равна суммарной производительности его портов:

$Ck = \sum Cpi.$

Первый коммутатор для локальных сетей не случайно появился для технологии Ethernet. Кроме очевидной причины, связанной с наибольшей популярностью сетей Ethernet, существовала и другая, не менее важная причина — эта технология больше других страдает от повышения времени ожидания доступа к среде при повышении загрузки сегмента. Поэтому сегменты Ethernet в крупных сетях в первую очередь нуждались в средстве разгрузки узких мест сети, и этим средством стали коммутаторы.

Некоторые компании стали развивать технологию коммутации для повышения производительности других технологий локальных сетей, таких как Token Ring и FDDI. Эти коммутаторы поддерживали как

алгоритм работы прозрачного моста, так и алгоритм моста с маршрутизацией от источника. Внутренняя организация коммутаторов различных производителей иногда очень отличалась от структуры первого коммутатора EtherSwitch, однако принцип параллельной обработки кадров по каждому порту оставался неизменным.

Широкому применению коммутаторов, безусловно, способствовало то обстоятельство, что внедрение технологии коммутации не требовало замены установленного в сетях оборудования — сетевых адаптеров, концентраторов, кабельной системы. Порты коммутаторов работали в обычном полудуплексном режиме, поэтому к ним прозрачно можно было подключить как конечный узел, так и концентратор, организующий целый логический сегмент.

Так как коммутаторы и мосты прозрачны для протоколов сетевого уровня, то их появление в сети не оказало никакого влияния на маршрутизаторы сети, если они там имелись.

Удобство использования коммутатора состоит еще и в том, что это самообучающееся устройство и, если администратор не нагружает его дополнительными функциями, конфигурировать его не обязательно — нужно только правильно подключить разъемы кабелей к портам коммутатора, а дальше он будет работать самостоятельно и эффективно выполнять поставленную перед ним задачу повышения производительности сети.

6.4. Полнодуплексные протоколы локальных сетей

Изменения в работе MAC-уровня при полнодуплексной работе

Технология коммутации сама по себе не имеет непосредственного отношения к методу доступа к среде, который используется портами коммутатора. При подключении сегментов, представляющих собой разделяемую среду, порт коммутатора должен поддерживать полудуплексный режим, так как является одним из узлов этого сегмента.

Однако, когда к каждому порту коммутатора подключен не сегмент, а только один компьютер, причем по двум раздельным каналам, как это происходит почти во всех стандартах физического уровня, кроме коаксиальных версий Ethernet, ситуация становится не такой однозначной. Порт может работать как в обычном полудуплексном режиме, так и в полнодуплексном. Подключение к портам коммутатора не сегментов, а отдельных компьютеров называется *микросегментацией.*

В полудуплексном режиме работы порт коммутатора по-прежнему распознает коллизии. Доменом коллизий в этом случае будет участок сети, включающий передатчик коммутатора, приемник коммутатора, передатчик сетевого адаптера компьютера, приемник сетевого адаптера компьютера и две витые пары, соединяющие передатчики с приемниками (рис. 6.12).

Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров, считая, что изображенный на рисунке сегмент свободен. Правда, вероятность коллизии в таком сегменте гораздо меньше, чем в сегменте, состоящем из 20-30 узлов, но она не нулевая. При этом максимальная производительность сегмента Ethernet в 14 880 кадров в секунду при минимальной длине кадра делится между передатчиком порта коммутатора и передатчиком сетевого адаптера. Если считать, что она делится пополам, то каждому предоставляется возможность передавать примерно по 7440 кадров в секунду.

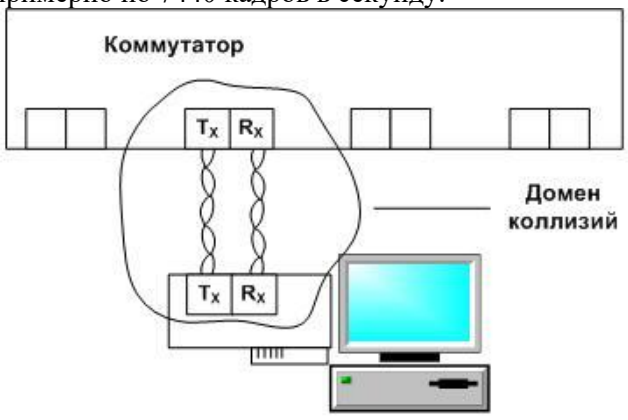


Рисунок 6.12. Домен коллизий, образуемый компьютером и портом коммутатора

В полнодуплексном режиме одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для индивидуальных полнодуплексных каналов связи, и он часто используется в протоколах территориальных сетей. При полнодуплексной связи порты Ethernet могут передавать данные со скоростью 20 Мбит/с — по 10 Мбит/с в каждом направлении. В случае Fast Ethernet и Gigabit Ethernet

так же имеем двукратную разницу в скорости работы порта в зависимости от режима.

Естественно, необходимо, чтобы MAC-узлы взаимодействующих устройств поддерживали этот специальный режим. В случае когда только один узел будет поддерживать полнодуплексный режим, второй узел будет постоянно фиксировать коллизии и приостанавливать свою работу, в то время как другой узел будет продолжать передавать данные, которые никто в этот момент не принимает. Изменения, которые нужно сделать в логике MAC-узла, чтобы он мог работать в полнодуплексном режиме, минимальны — нужно просто отменить фиксацию и отработку коллизий в сетях Ethernet, а в сетях Token Ring и FDDI — посылать кадры в коммутатор, не дожидаясь прихода токена доступа, а тогда, когда это нужно конечному узлу. Фактически, при работе в полнодуплексном режиме MAC-узел не использует метод доступа к среде, разработанный для данной технологии.

Так как переход на полнодуплексный режим работы требовал изменения логики работы MAC-узлов и драйверов сетевых адаптеров, то он сначала был опробован при соединении двух коммутаторов. Уже первые модели коммутатора EtherSwitch компании Kalpana поддерживали полнодуплексный режим при взаимном соединении, обеспечивая скорость взаимного обмена 20 Мбит/с.

Позже появились версии полнодуплексного соединения FDDI-коммутаторов, которые при одновременном использовании двух колец FDDI обеспечивали скорость обмена в 200 Мбит/с.

После опробования полнодуплексной технологии на соединениях коммутатор-коммутатор разработчики реализовали ее и в сетевых адаптерах Ethernet и Fast Ethernet. При разработке технологий Fast Ethernet и Gigabit Ethernet полнодуплексный режим стал одним из двух полноправных стандартных режимов работы узлов сети. Все сетевые адаптеры сейчас могут поддерживать оба режима работы, отрабатывая логику алгоритма доступа CSMA/CD при подключении к порту концентратора и работая в полнодуплексном режиме при подключении к порту коммутатора.

При использовании полнодуплексных версий протоколов происходит некоторое сближение различных технологий, так как метод доступа во многом определял лицо каждой технологии. Различие технологий остается в различных форматах кадров, а также в процедурах контроля корректности работы сети на физическом и канальном уровнях.

Полнодуплексные версии протоколов могли бы быть реализованы и в мостах. Принципиальных препятствий для этого не было, просто в период применения локальных мостов потребности в высокоскоростной передаче межсегментного трафика не возникало.

Проблема управления потоком данных при полнодуплексной работе

Простой отказ от поддержки алгоритма доступа к разделяемой среде без какой-либо модификации протокола ведет к повышению вероятности потерь кадров коммутаторами, так как при этом теряется контроль за потоками кадров, направляемых конечными узлами в сеть. Раньше поток кадров регулировался методом доступа к разделяемой среде, так что слишком часто генерирующий кадры узел вынужден был ждать своей очереди к среде и фактическая интенсивность потока данных, который направлял в сеть этот узел, была заметно меньше той интенсивности, которую узел хотел бы отправить в сеть. При переходе на полнодуплексный режим узлу разрешается отправлять кадры в коммутатор всегда, когда это ему нужно, поэтому коммутаторы сети могут в этом режиме сталкиваться с перегрузками, не имея при этом никаких средств регулирования («притормаживания») потока кадров.

Причина перегрузок обычно кроется не в том, что коммутатор является блокирующим, то есть ему не хватает производительности процессоров для обслуживания потоков кадров, а в ограниченной пропускной способности отдельного порта, которая определяется временными параметрами протокола. Например, порт Ethernet не может передавать больше 14 880 кадров в секунду, если он не нарушает временных соотношений, установленных стандартом.

Поэтому, если входной трафик неравномерно распределяется между выходными портами, легко представить ситуацию, когда в какой-либо выходной порт коммутатора будет направляться трафик с суммарной средней интенсивностью большей, чем протокольный максимум. На рис. 6.13 изображена как раз такая ситуация, когда в порт *3* коммутатора направляется трафик от портов *1,2, 4 и 6, с* суммарной интенсивностью в 22 100 кадров в секунду. Порт *3* оказывается загружен на 150 %. Естественно, что когда кадры поступают в буфер порта со скоростью 20 100 кадров в секунду, а уходят со скоростью 14 880 кадров в секунду, то внутренний буфер выходного порта начинает неуклонно заполняться необработанными кадрами.

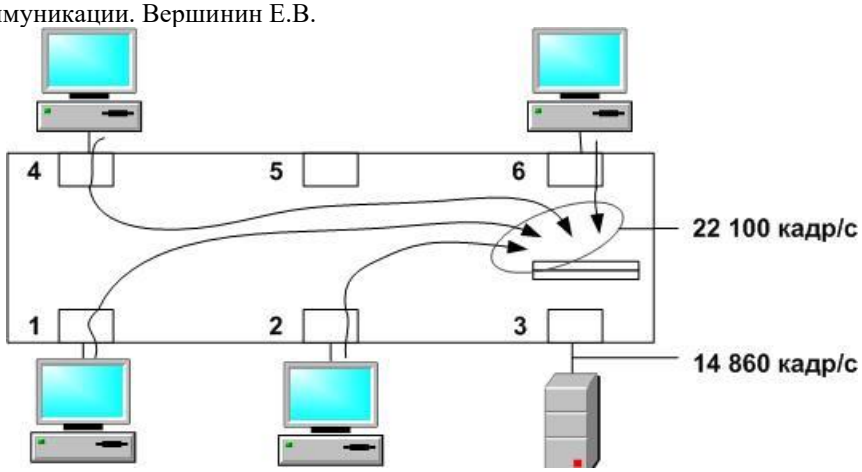


Рисунок 6.13. Переполнение буфера при несбалансированном трафике

Какой бы ни был объем буфера порта, он в какой-то момент времени обязательно переполнится. Нетрудно подсчитать, что при размере буфера в 100 Кбайт в приведенном примере полное заполнение буфера произойдет через 0,22 секунды после начала его работы (буфер такого размера может хранить до 1600 кадров размером в 64 байт). Увеличение буфера до 1 Мбайт даст увеличение времени заполнения буфера до 2,2 секунд, что также неприемлемо. А потери кадров всегда очень нежелательны, так как снижают полезную производительность сети, и коммутатор, теряющий кадры, может значительно ухудшить производительность сети вместо ее улучшения.

В первых глобальных сетях использовались коммутаторы технологии X.25, которые поддерживали протокол канального уровня LAP-B. Он имел специальные кадры управления потоком «Приемник готов» (RR) и «Приемник не готов» (RNR), аналогичные по назначению кадрам протокола LLC2 (это не удивительно, так как оба протокола принадлежали семейству протоколов HDLC). Протокол LAP-B работал между соседними коммутаторами сети X.25 и в том случае, когда очередь коммутатора доходила до опасной границы, запрещал своим ближайшим соседям с помощью кадра «Приемник не готов» передавать ему кадры, пока очередь не уменьшится до нормального уровня. В сетях X.25 такой протокол был необходим, так как эти сети никогда не использовали разделяемые среды передачи данных, а работали по индивидуальным каналам связи в полнодуплексном режиме.

При разработке коммутаторов локальных сетей ситуация коренным образом отличалась от ситуации, при которой создавались коммутаторы территориальных сетей. Основной задачей было сохранение конечных узлов в неизменном виде, что исключало корректировку протоколов локальных сетей. В этих протоколах процедур управления потоком не было — общая среда передачи данных в режиме разделения времени исключала возникновение ситуаций, когда сеть переполнялась бы необработанными кадрами. Сеть не накапливала данных в каких-либо промежуточных буферах при использовании только повторителей или концентраторов.

Применение коммутаторов без изменения протокола работы оборудования всегда порождает опасность потери кадров. Если порты коммутатора работают в обычном, то есть в полудуплексном режиме, то у коммутатора имеется возможность оказать некоторое воздействие на конечный узел и заставить его приостановить передачу кадров, пока у коммутатора не разгрузятся внутренние буферы.

Если же коммутатор работает в полнодуплексном режиме, то протокол работы конечных узлов, да и его портов все равно меняется. Поэтому имело смысл для поддержки полнодуплексного режима работы коммутаторов несколько модифицировать протокол взаимодействия узлов, встроив в него явный механизм управления потоком кадров.

Работа над выработкой стандарта для управления потоком кадров в полнодуплексных версиях Ethernet и Fast Ethernet продолжалась несколько лет. Такой длительный период объясняется разногласиями членов соответствующих комитетов по стандартизации, отстаивающих подходы фирм, которые реализовали в своих коммутаторах собственные методы управления потоком.

В марте 1997 года принят стандарт IEEE 802.3x на управление потоком в полнодуплексных версиях протокола Ethernet. Он определяет весьма простую процедуру управления потоком, подобную той, которая используется в протоколах LLC2 и LAP-B. Эта процедура подразумевает две команды — «Приостановить передачу» и «Возобновить передачу», которые направляются соседнему узлу. Отличие от протоколов типа LLC2 в том, что эти команды реализуются на уровне символов кодов физического уровня, таких как 4B/5B, а не на уровне команд, оформленных в специальные управляющие кадры. Сетевой адаптер или порт коммутатора, поддерживающий стандарт 802.3x и получивший команду «Приостановить передачу», должен прекратить передавать кадры впредь до получения команды

«Возобновить передачу».

Такая простая процедура управления потоком оказалась плохо пригодной в сетях Gigabit Ethernet. Полная приостановка приема кадров от соседа при такой большой скорости передачи кадров (1 488 090 кадр/с) может быстро вызвать переполнение внутреннего буфера у этого предыдущего соседа, который в свою очередь полностью заблокирует прием кадров у своих ближайших соседей. Таким образом, перегрузка просто распространится по сети, вместо того чтобы постепенно исчезнуть. Для работы с такими скоростными протоколами необходим более тонкий механизм регулирования потока, который бы указывал, на какую величину нужно уменьшить интенсивность потока входящих кадров в перегруженный коммутатор, а не приостанавливал этот поток до нуля. Подобный плавный механизм регулирования потока появился у коммутаторов ATM через несколько лет после их появления.

Ограничения мостов и коммутаторов

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

• Во-первых, в топологии получившейся сети должны *отсутствовать петли.* Действительно, мост/коммутатор может решать задачу доставки пакета адресату только тогда, когда между отправителем и получателем существует единственный путь. В то же время наличие избыточных связей, которые и образуют петли, часто необходимо для лучшей балансировки нагрузки, а также для повышения надежности сети за счет образования резервных путей.

• Во-вторых, логические сегменты сети, расположенные между мостами или коммутаторами, *слабо изолированы* друг от друга, а именно не защищены от так называемых широковещательных штормов. Если какая-либо станция посылает широковещательное сообщение, то это сообщение передается всем станциям всех логических сегментов сети. Защита от широковещательных штормов в сетях, построенных на основе мостов и коммутаторов, имеет количественный, а не качественный характер: администратор просто ограничивает количество широковещательных пакетов, которое разрешается генерировать некоторому узлу в единицу времени. Использование же механизма виртуальных сетей, реализованного во многих коммутаторах, хотя и позволяет достаточно гибко создавать изолированные по трафику группы станций, но при этом изолирует их полностью, так что узлы одной виртуальной сети не могут взаимодействовать с узлами другой виртуальной сети.

• В-третьих, в сетях, построенных на основе мостов и коммутаторов, достаточно сложно решается задача управления трафиком на основе значения данных, содержащихся в пакете. В таких сетях это возможно только с помощью пользовательских фильтров, для задания которых администратору приходится иметь дело с двоичным представлением содержимого пакетов.

• В-четвертых, реализация транспортной подсистемы только средствами физического и канального уровней, к которым относятся мосты и коммутаторы, приводит к недостаточно гибкой, одноуровневой системе адресации: в качестве адреса назначения используется MAC-адрес, жестко связанный с сетевым адаптером.

• Наконец, возможностью трансляции протоколов канального уровня обладают далеко не все типы мостов и коммутаторов, к тому же эти возможности ограничены. В частности, в объединяемых сетях должны совпадать максимально допустимые размеры полей данных в кадрах, так как мостами и коммутаторами не поддерживается функция фрагментации кадров.

Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях — это привлечение средств более высокого, сетевого уровня.

Выводы

- Логическая структуризация сети необходима при построении сетей средних и крупных размеров.
- Деление сети на логические сегменты повышает производительность, надежность, гибкость построения и управляемость сети.
- Для логической структуризации сети применяются коммутаторы и маршрутизаторы. Первые позволяют разделить сеть на логические сегменты с помощью минимума средств — только на основе протоколов канального уровня. Кроме того, эти устройства не требуют конфигурирования.

- Логические сегменты, построенные на основе коммутаторов, являются строительными элементами более крупных сетей, объединяемых маршрутизаторами.
- Коммутаторы — наиболее быстродействующие современные коммуникационные устройства, они позволяют соединять высокоскоростные сегменты без блокирования (уменьшения пропускной способности) межсегментного трафика.
- Пассивный способ построения адресной таблицы коммутаторами — с помощью слежения за проходящим трафиком — приводит к невозможности работы в сетях с петлевидными связями. Другим недостатком сетей, построенных на коммутаторах, является отсутствие защиты от широковещательного шторма, который эти устройства обязаны передавать в соответствии с алгоритмом работы.
- Применение коммутаторов позволяет сетевым адаптерам использовать полнодуплексный режим работы протоколов локальных сетей (Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI). В этом режиме отсутствует этап доступа к разделяемой среде, а общая скорость передачи данных удваивается.
- В полнодуплексном режиме для борьбы с перегрузками коммутаторов используется метод управления потоком, описанный в стандарте 802.3x. Он повторяет алгоритмы полной приостановки трафика по специальной команде, известной из технологий глобальных сетей.