



# КОММУТАЦИЯ

В локальных сетях

Базовый

Сетевые технологии  
и системное администрирование

# Урок №4

## Коммутация в локальных сетях. Базовый

### Содержание

<b>1. Логическая структуризация сети .....</b>	<b>4</b>
<b>2. Структуризация сети с применением мостов и коммутаторов .....</b>	<b>9</b>
<b>3. История появления FastEthernet .....</b>	<b>38</b>
<b>4. Физический уровень технологии FastEthernet .....</b>	<b>45</b>
<b>5. Методика расчета конфигурации сети FastEthernet .....</b>	<b>54</b>
<b>6. Технология GigabitEthernet (802.3Z).....</b>	<b>57</b>
<b>7. Физические реализации Gigabit Ethernet .....</b>	<b>60</b>
<b>8. 10 GigabitEthernet и быстрее.....</b>	<b>64</b>
<b>Приложение .....</b>	<b>68</b>

## ***Уважаемые студенты!***

*На прошлом уроке мы познакомились с принципами работы технологии Ethernet, которая появилась в начале 80-х годов 20 века и прочно заняла место на рынке как основная технология построения недорогих локальных сетей. Мы познакомились с принципами работы канального уровня Ethernet, узнали, как устроен метод доступа, что такое коллизии, как должны обрабатываться коллизии и какие ограничения накладывает на сети выбранный метод доступа, познакомились с форматом кадра Ethernet и реализациями физического уровня.*

*Ясно, что сегодня локальная сеть с пропускной способностью 10 Мбит/с не может удовлетворять современных потребностей в передаче данных в локальных сетях и применение Ethernet в том виде, в котором мы его изучили невозможно. Однако, мы потратили время не зря: новые технологии, которые мы применяем и сегодня, выросли из Ethernet и глубокое знание исходной реализации Ethernet все так же актуально и сегодня.*

*Теперь мы продолжим изучать, как развивалась технология Ethernet вместе с другими решениями IEEE 802 и обсудим в этом уроке две важнейшие темы: логическая структуризация сетей и высокоскоростные реализации Ethernet.*

# 1. Логическая структуризация сети

---

25 лет назад для построения простейшей локальной сети достаточно было, помимо, разумеется, линий связи, наличия только сетевых адаптеров и концентраторов. Но при росте размеров сетей даже дополнительные возможности сетевых концентраторов уже не могут справиться с возникающими проблемами, и проектировщик сталкивается с необходимостью логической структуризации сети. Давайте обсудим, что же это такое.

Мы с вами отмечали, что наличие коллизий — это нормальное явление для сетей технологии Ethernet, которая использует метод CSMA/CD. Также определили такое понятие, как домен коллизий. Тогда мы говорили, что сети Ethernet — это по сути один домен коллизий.

В одном домене коллизий все станции сети получают абсолютно все кадры и обнаруживают коллизии. Также алгоритмом обработки коллизий было установлено предельное число станций в сети или, лучше сказать, в одном домене коллизий — 1024, и чем ближе мы подбирались к этой цифре, тем хуже становилась производительность (мы говорили, что это ограничение совершенно не актуально сегодня, просто следует говорить о том, что сеть должна быть недогружена). Однако есть возможность логически организовать разделение сети на несколько доменов коллизий, при этом один сегмент сети не будет «видеть» коллизий, которые возникают в другом сегменте.

Если учитывать тот факт, что увеличение интенсивности коллизий приводит к уменьшению пропускной способности сети, то такой способ структуризации должен позволить очень эффективно решить проблемы, связанные с особенностями метода доступа Ethernet.

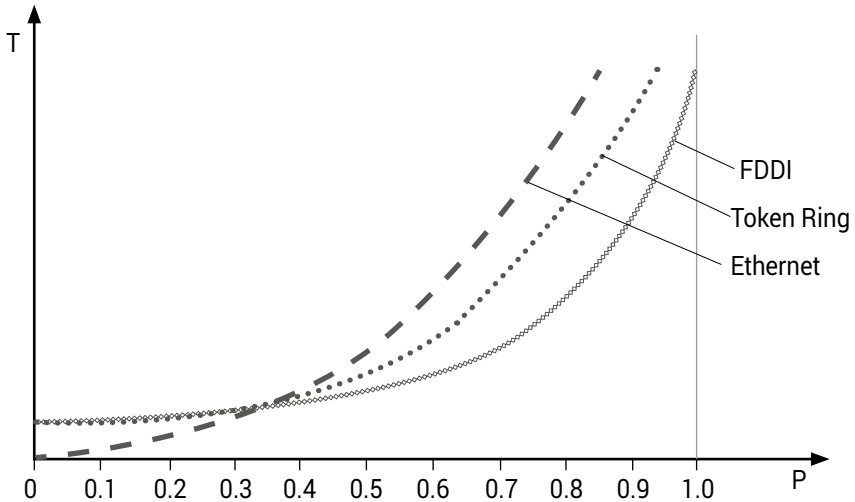
Итак, логическая структуризация сети — это разбиение с помощью специального оборудования — мостов, коммутаторов или маршрутизаторов — общей разделяемой среды на логические сегменты, каждый из которых представляет собой отдельную разделяемую среду с меньшим количеством узлов. Результатом становится улучшение производительности и надежности.

Давайте еще раз подчеркнем, в чем же состоит основная причина необходимости разбивать сеть на относительно независимые логические сегменты? Хотя разделяемая среда и является достаточно эффективным средством для построения небольшой сети, при попытке превысить некий порог в числе станций, или, лучше сказать, в нагрузке на сеть, мы неминуемо сталкиваемся с катастрофическим падением производительности.

Это очень ярко проявляется в технологии Ethernet. Нельзя гарантировать для сегмента 10 Мбит/с Ethernet с количеством узлов  $N$ , что конкретному узлу достанется  $10/N$  часть пропускной способности. Более того, нельзя даже гарантировать, что этому узлу вообще удастся получить доступ к среде. Причина этого заключается в случайном характере метода доступа CSMA/CD.

Для наглядности приведем зависимость задержек доступа к среде передачи данных в сетях Ethernet, Token Ring и FDDI от коэффициента использования сети, ко-

торый часто называют коэффициентом нагрузки сети (коэффициент использования сети равен отношению трафика, который должна передать сеть, к ее максимальной пропускной способности).



Как видно из рисунка, все технологии характеризуются экспоненциальным ростом 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 технологии FDDI это 70%, для технологии Token Ring — 60%, а для Ethernet — 40–50%.

При загрузке сети до этого порога, сеть Ethernet на разделяемом сегменте хорошо справляется с передачей трафика, генерируемого конечными узлами. Однако при повышении его интенсивности, все больше времени уделяется повторной передаче кадров, вызвавших кол-

лизии, и при дальнейшем возрастании интенсивности генерируемого трафика, вероятность столкновения кадров настолько увеличивается, что практически любой кадр, который какая-либо станция пытается передать, сталкивается с другими кадрами, вызывая коллизию.

Теперь представьте себе, что вместо построения сети, которая вся будет одним большим доменом коллизий, нам удалось построить сеть, состоящую из нескольких независимых доменов коллизий, разделенных некоторыми специальными устройствами. Ясно, что суммарная пропускная способность такой сети больше, а задержки доступа к среде в такой сети меньше.

В качестве интеллектуальных устройств, с помощью которых производят логическую структуризацию сети можно выделить устройства канального уровня (мосты, коммутаторы) и сетевого уровня (маршрутизаторы). Мосты и коммутаторы позволяют, сохранив все устройства в одной локальной сети, (т. е. сохранив возможность обмениваться кадрами канального уровня так, как это происходит в обычной, неструктурированной сети), просто разделить эту одну канальную сеть на домены коллизий. Маршрутизаторы позволяют соединить несколько различных канальных сетей в одну большую составную сеть. Устройства составной сети не могут непосредственно обмениваться кадрами канального уровня, они обмениваются пакетами сетевого уровня, которые в каждой канальной сети снабжаются необходимыми заголовками канального уровня для передачи по данной канальной сети.

В современных сетях часто используют комбиниро-

ванный способ логической структуризации: небольшие сегменты объединяются устройствами канального уровня в более крупные сети, которые, в свою очередь, соединяются маршрутизаторами на сетевом уровне. Изучение работы маршрутизаторов нам предстоит в следующих курсах, а пока мы познакомимся с принципами работы устройств канального уровня: мостов и коммутаторов, благодаря которым можно производить сегментацию сети на канальном уровне, т. е. разбивать канальную сеть на сегменты так, что все устройства по-прежнему находятся в одной канальной сети, но при этом каждый сегмент относительно независим и является самостоятельным доменом коллизий.

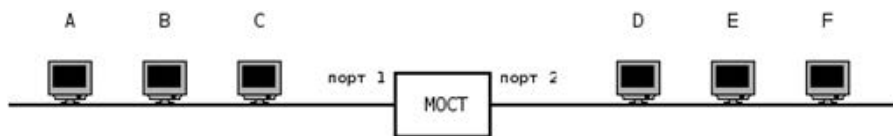


## 2. Структуризация сети с применением мостов и коммутаторов

---

Давайте рассмотрим принцип работы моста. Положим, у нас была простая сеть 10Base2, состоящая из двух сегментов с рабочими станциями, сегменты были соединены повторителем.

Эта сеть, как мы знаем, представляла собой один большой домен коллизий, так как сигнал, посланный любой станцией достигал ВСЕХ станций сети, т. е. всех станций в обоих сегментах и каждая пара станций в этой сети могла вызвать коллизию. Почему это было возможно? Потому, что повторитель — не интеллектуальное устройство, получая сигнал из правого сегмента, оно передает его в левый сегмент совершенно не интересуясь тем, что, возможно, левый сегмент УЖЕ занят чьей-то передачей, и если в левом сегменте в настоящее время среда занята некоторой станцией, то повторитель передаст в левый сегмент кадр, поступивший из правого и вызовет коллизию. Фактически, повторитель в этом вопросе не интеллектуальнее кабеля, он работает на физическом уровне и передает сигналы совершенно не заботясь о том, что это вызовет коллизию в выходном сегменте. Теперь давайте представим, что вместо повторителя мы используем более «умное» устройство, которое, прежде чем передать сигнал в выходной порт, проверяет, не занята ли там среда.



Как проверяет? Точно так же как, обычная станция перед тем, как передать данные! Ведь прежде, чем передавать имеющийся у станции в буферной памяти кадр в сеть, станция СЛУШАЕТ сеть, и только убедившись в том, что она свободна, начинает передачу своего кадра, иными словами станция выполняет известный нам метод доступа CSMA/CD. А что если наше «умное» устройство будет поступать точно так же?

Положим «умное» устройство получило из левого сегмента некоторый кадр (сигнал, пока не важно, как это называть). Умное устройство, в отличие от повторителя, который просто ретранслирует кадр в правый сегмент, сперва проверяет, не занят ли правый сегмент так, как делала бы станция перед передачей кадра и выдает кадр в правый сегмент только если среда там свободна, если же среда занята, то «умное» устройство ожидает, пока среда правого сегмента освободиться и только потом начинает выдачу в правый сегмент. Что нам даст такое поведение? Оно даст нам очень много! Теперь левый и правый сегмент независимы с точки зрения коллизий! Действительно, если ЛЮБАЯ передача данных из левого сегмента не попадет в правый пока среда там занята, это означает, что станция левого сегмента не может породить коллизию в паре со станцией правого сегмента, следовательно левый и правый сегмент — независимые домены

коллизий. Это дает нам огромный шаг вперед, так как позволяет преодолеть ограничения CSMA/CD на размер сети. Действительно, сеть Ethernet имеет жесткие ограничения на длину, связанные не с физическим уровнем (проблемы физического уровня, а именно затухание и иное повреждение сигнала устраняется повторителями), а с канальным уровнем, именно необходимость распознавания коллизий ограничивает количество применяемых повторителей и, в итоге, общую длину классической сети Ethernet. И теперь, с использованием описанного «умного» устройства, мы можем соединить две сети Ethernet (каждая — максимального размера с точки зрения PDV/PVV) и такая большая сеть будет вполне корректно работать, ведь коллизии теперь не могут происходить между станциями различных «половинок» большой сети, только внутри каждой «половинки», а «половинки», как мы оговорили изначально, работают корректно, следовательно и вся получившаяся сеть тоже корректна. Ясно, что применение такого «умного» устройства, позволяя нам разделить сеть на независимые домены коллизий, фактически снимает ограничения на общую длину сети Ethernet, такие «умные» устройства называют МОСТ.

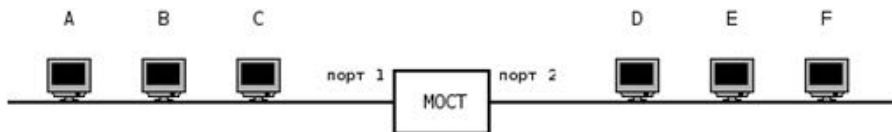
Однако пока что мы обсудили лишь самую базовую идею, лежащую в основе моста, давайте выясним, как работают мосты более детально. Для начала давайте обсудим вот что. Положим, в левом сегменте начинается передача данных, эти данные нужно передать в правый сегмент, а он пока занят другим кадром, мы говорили, что мост должен ПОДОЖДАТЬ и начать передавать кадр в правый сегмент только когда среда в нем освободится. Это озна-

чает, что мост, получая кадр слева, должен разместить его в специальной буферной памяти, в которой кадр будет дожидаться момента, пока среда справа освободится. У повторителя нет такой памяти, он просто передает сигналы, а мосты, очевидно, должны иметь буферную память для каждого своего порта (или общую память для обоих портов, пока не столь важно).

Итак, мы сделали первый важный вывод: мост это устройство, которое, в отличие от концентратора имеет память и может буферизировать полученные данные, которые пока нельзя выдать в выходной порт.

Давайте пойдем дальше: можно ли буферизировать полученный СИГНАЛ, чтобы затем, когда среда освободится, передать его в выходной порт? Технически, конечно, можно, но получая цифровой сигнал, его гораздо разумнее представить как совокупность бит или байт и хранить в буферной памяти в виде кадра, готового к отправке. Таким образом мы делаем второй вывод: мост оперирует не просто сигналами, но кадрами, т. е. работает не только на физическом, но и на канальном уровне. Однако не только тот факт, что мост оперирует кадрами говорит нам о том, что он работает на канальном уровне, каждый порт моста, как уже говорились выше, выполняет всю логику CSMA/CD, следовательно, снова таки, работает на канальном уровне.

Давайте рассмотрим пример, подчеркивающий, что мост полноценно выполняет логику CSMA/CD на своих портах. Положим, мы соединили мостом два сегмента 10Base2.



Положим, станция A передает кадр станции B, и в это время станция D начинает передачу своего кадра, положим для станции C. Кадр от станции D достигает второго порта моста и должен быть передан на первый порт, однако среда слева пока занята. Как мы уже сказали, мост буферизирует этот кадр и ждет, пока среда слева станет свободна чтобы затем передать кадр станции D в сегмент слева. Что это означает?

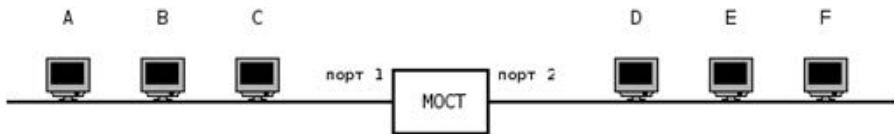
Мост на порту 1 ведет себя в точности как станция, у которой есть кадр для передачи в среду: он ждет пока среда слева освободится, потом ждет IPG и пытается начать свою передачу в левый сегмент. А что, если после того, как мост подождет IPG, он выяснит, что среда снова занята? Это вполне возможно, например потому, что станция B начала передачу своего кадра несколько раньше, чем это начал делать мост. Тогда мост, как любая другая станция в CSMA/CD среде подождет, пока среда снова станет свободной, затем подождет IPG и снова попытает начать свою передачу кадра от станции D в левый сегмент.

Положим, мост начал такую передачу, но почти одновременно с ним станция C тоже начала свою передачу, это вполне возможная ситуация в разделяемой среде. Произойдет коллизия, распознавать коллизия должны те, кто ее создал, т. е. станция C и порт 1 моста. После того, как коллизия обработана, станция C и порт 1 моста выдержат IPG, затем случайную паузу (каждый — свою) и снова

попытаются повторно передать свои кадры из выходного буфера. Как видите, порт моста с точки зрения CSMA/CD ведет себя полностью как обычный узел, борясь за среду, детектируя коллизии и выполняя повторные передачи кадров, которые не удалось передать из-за коллизии.

Итак, мост это устройство канального уровня, которое разделяет сеть на домены коллизий, выполняя на каждом своем порту полноценную логику CSMA/CD и имеет буферную память для хранения кадров с данными.

Теперь давайте обсудим еще одну очень важную вещь, снова обратимся к простой топологии, используемой нами выше:



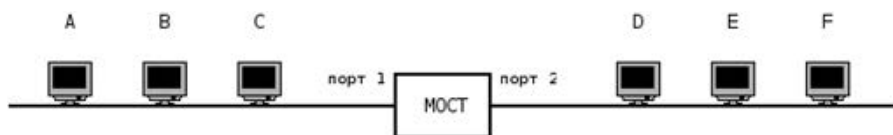
В прошлом примере мы рассматривали ситуацию, в которой станция A передает кадр станции B, а в это время среда справа свободна и станция D передает свой кадр. Не заметили ли вы здесь подвоха? Как среда справа может быть свободна, если станция A передает кадр станции B? Ведь мост должен был бы, как и повторитель, передать этот кадр и в правый сегмент, что привело бы к занятости среды справа и невозможности станции D передавать свой кадр до тех пор, пока станция A передает свой?

Как сказали выше, мост — устройство канального, а не физического уровня, он работает с кадрами и анализирует их содержимое, поэтому мост, получая кадр, может не только его буферизировать при необходимости, но и анализировать, например, выяснить, какой станции (т. е.

какому MAC адресу) предназначен данный кадр. Давайте представим, что у моста есть (пока не важно откуда) таблица, в которой указано, какие MAC адреса находятся за какими портами моста. В таком случае мост, получая слева кадр от станции А для станции В видит, что станция получатель этого кадра находится за тем же портом, откуда кадр поступил, поэтому данный кадр не нужно перенаправлять в правый сегмент сети, более того, его не нужно перенаправлять и в левый сегмент, так как в левом сегменте такой кадр уже есть (он оттуда поступил) и поэтому если получатель имел возможность получить данный кадр, он его уже получил. Мосту необходимо просто отбросить (отфильтровать) данный кадр после получения, таким образом, пока среда слева занята передачей кадра от А к В, среда слева — свободна и станция D действительно может отправить свой кадр к станции В. Получая такой кадр мост видит, что получатель кадра за первым портом и принимает решение о передаче (форвардинге) этого кадра через первый порт, а так как среда слева занята, то, в соответствии с рассмотренным выше, мост буферизирует данный кадр и начинает бороться за доступ к среде в левом сегменте.

Теперь давайте рассмотрим, откуда мост берет упомянутую таблицу соответствий MAC адресов и портов, за которыми эти адреса находятся. Первое, что приходит в голову — такую таблицу может заполнять администратор вручную. Ясно, что это не самое лучшее решение: администратору придется выполнять большой объем ручной работы, при ручном заполнении таблицы легко ошибиться, так как MAC адреса не имеют структуры и

их сложно запоминать, наконец, узлы могут выключаться, добавляться и даже перемещаться в сети. И хотя внесение вручную записей в таблицу нередко возможно, это — не основной способ заполнения таблицы соответствий MAC адресов и портов. Давайте рассмотрим, как эта таблица типично заполняется.



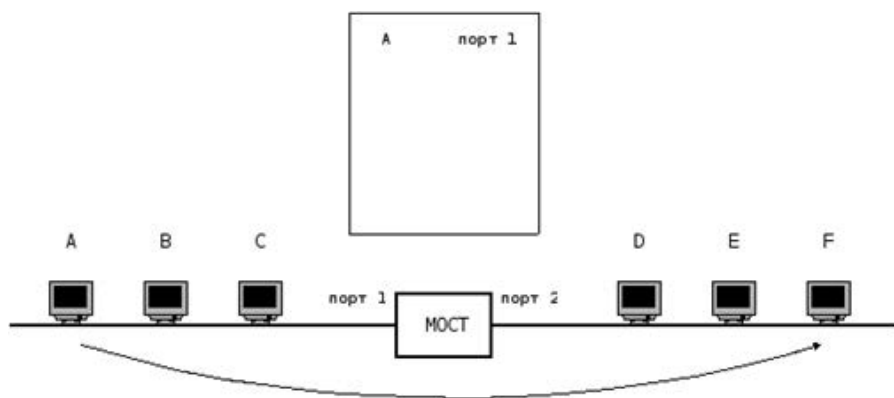
Сразу после запуска моста, его таблица соответствий MAC адресов и портов предсказуемо пуста. Положим, на мост поступил кадр от станции А для станции В. Так как мост не знает, где находится станция В (за первым или за вторым портом), он должен НЕ НАВРЕДИТЬ, т. е. поступить не хуже, чем поступил бы на его месте концентратор, поэтому в этой ситуации мост ПЕРЕДАЕТ кадр от станции А, предназначенный для станции В через свой второй порт.

Такой способ обработки кадров с неизвестным адресом получателя называется flooding (затопление). Да, этот кадр не нужен за вторым портом, но мост этого пока не знает и должен вести себя так, чтобы не навредить. После того, как мост передал кадр от станции А на второй порт он делает очень важную вещь, которая и лежит в основе самообучения моста: он анализирует поле SA в заголовке Ethernet принятого кадра и делает вывод о том, что раз кадр с данным MAC адресом отправителя поступил на первый порт, то обладатель этого MAC адреса находится



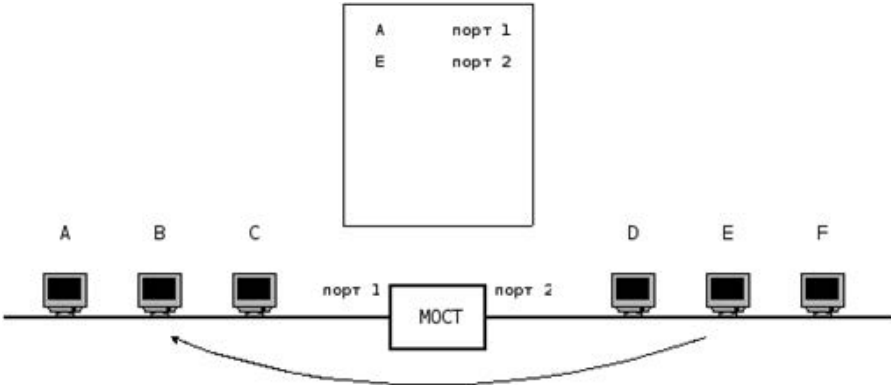
за первым портом, после чего он записывает в своей таблице соответствий MAC адресов и портов, что станция А находится за первым портом. Обратите внимание, мост не может из данного кадра узнать, где находится получатель, станция В, он лишь может узнать, за каким портом находится отправитель кадра, станция А.

Однако, вернемся к нашему примеру. Предположим, теперь станция с MAC адресом А отправила кадр, предназначенный для станции с MAC адресом F. Мост, не зная пока, где находится станция F, как и положено, передаст этот кадр на второй порт, при этом мост уже знает, за каким портом находится станция А, однако, тем не менее, каждый новый поступивший кадр используется для подтверждения актуальности этой информации (ведь местоположение станций в сети может меняться).

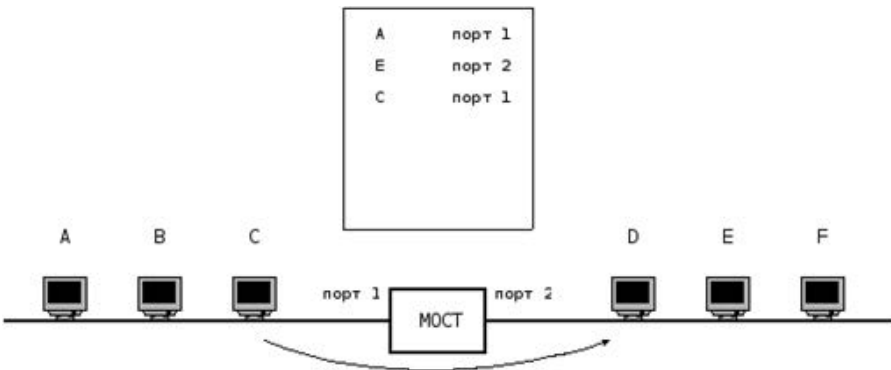


Далее, положим, станция с MAC адресом E отправляет кадр для станции с MAC адресом В. Данный кадр будет перенаправлен на первый порт, так как местоположение станции В еще не известно (и это хорошо, ведь станция В

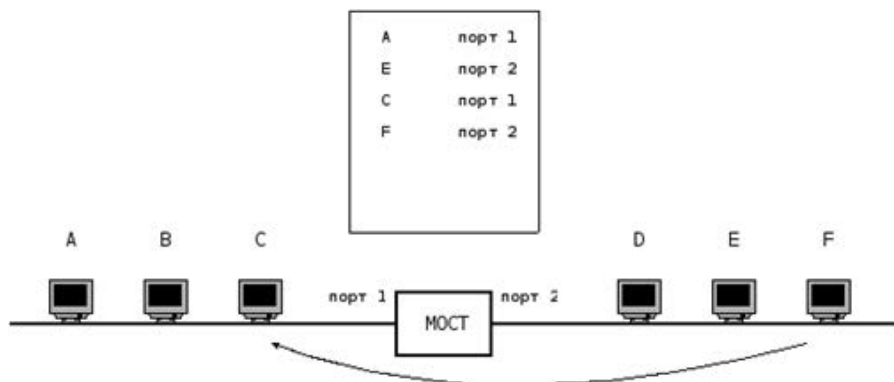
в другом сегменте), кроме того в таблице адресов моста появляется запись о местонахождении станции с MAC адресом E:



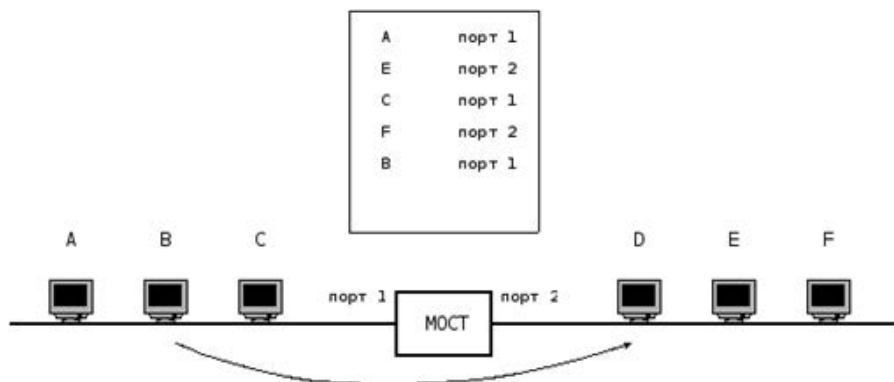
Пусть теперь станция с MAC адресом C отправляет кадр для станции с MAC адресом D. Данный кадр снова обрабатывается флуидингом, т. е. перенаправляется на второй порт (и это снова позволяет передать кадр станции D), а в таблице соответствий моста появляется запись о местонахождении станции с MAC адресом C:



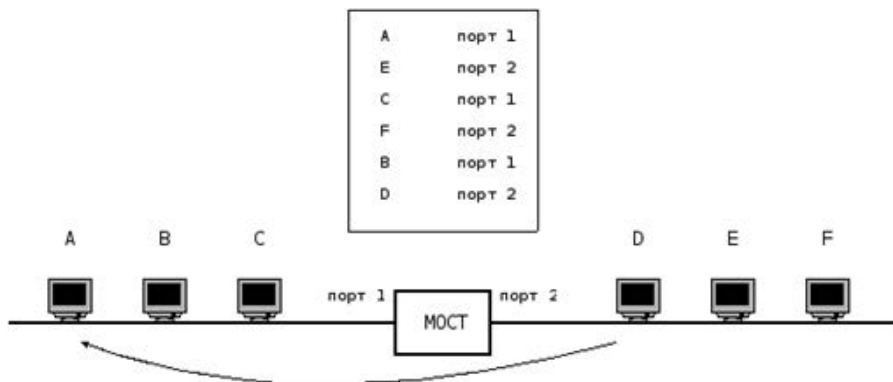
Затем станция с MAC адресом F отправляет кадр для станции с MAC адресом C, данный кадр перенаправляется на первый порт уже не в рамках флудинга, а сознательно, так как мост знает, что станция C находится за первым портом, а в таблице моста появляется запись о местонахождении станции с MAC адресом F:



Затем станция с MAC адресом B отправляет кадр для станции с MAC адресом D, здесь мост выполняет флудинг и в таблице моста появляется запись о местонахождении станции с MAC адресом B:



И, наконец, пусть станция с MAC адресом D отправляет кадр для станции с MAC адресом A, мост выполняет форвардинг, а в таблице моста появляется запись о местонахождении станции с MAC адресом D:



Теперь таблица заполнена. Процесс, называемый процессом самообучения моста завершен, и теперь можно пользоваться всеми преимуществами, предоставляемыми алгоритмом прозрачного моста. Теперь, если станция с MAC адресом A отправит кадр станции с MAC адресом F, то он все равно будет передан во второй сегмент через порт 2, а вот если кадр будет отправлен станции с MAC адресом C, находящейся в том же сегменте, мост определит, что выходной порт должен совпадать со входным и отфильтрует кадр. Еще раз: обратите внимание, что кадр будет уничтожен, а не отправлен в первый сегмент, ведь станция с MAC адресом C и порт первый моста находятся в одном сегменте, это означает, что станция-получатель уже получила этот кадр, и нет никакой необходимости в повторной его передаче через первый порт.

На самом деле процесс самообучения моста никогда не заканчивается, так как мост не знает, сколько станций находятся в каждом из сегментов. Кроме того, станции могут выходить из строя, новые станции могут быть добавлены в сегмент, существующие станции могут перемещаться из сегмента в сегмент, все эти изменения мост должен отслеживать и приспосабливаться к ним, поэтому, мост не ждет некоторого момента заполнения адресной таблицы, а начинает использовать новые записи по мере их возникновения.

Так как топология сети может изменяться вследствие перемещения станций или изменения их числа, записи в таблице соответствий моста имеют некий срок валидности, обычно несколько минут. Такие записи называются динамическими, и создаются в процессе самообучения моста. Периодически мост просматривает свою таблицу, и все записи, чей возраст старше оговоренного времени удаляются.

Таким образом, если станция некоторое время не проявляет активности, запись о ее MAC адресе удаляется, и предназначенные ей кадры вновь передаются мостом на все порты. Также записи могут вноситься в таблицу администратором, в этом случае они называются статическими, срока жизни такие записи не имеют.

В рассмотренных выше примерах мосты имели два порта, однако никакого принципиального ограничения на количество портов у устройства с описанной выше логикой, очевидно, нет. Мост мог иметь три и больше портов, за каждым из которых мог находиться сегмент Ethernet с любым физическим уровнем, например 10Base2/10Base5/10BaseT.

Само по себе слово «мост» предполагает, что соединяются ДВЕ сети, для многопортовых мостов применяется другой термин, такие устройства называют «коммутаторы». На самом деле это не совсем точно, давайте поговорим об этом чуть более детально.

Изначально мосты, хотя и были предназначены для увеличения производительности сети, были не слишком быстрыми устройствами. Так как мост в процессе работы должен выполнять описанную выше логику, т. е. анализировать заголовки, заполнять таблицу соответствий, искать для каждого поступившего кадра нужную строку в таблице соответствий, часто мост не мог работать на полной скорости Ethernet, т. е. не мог перенаправлять кадры из сегмента в сегмент со скоростью провода, т. е. на скорости 10 Мбит/с, например, мост мог передавать трафик из сегмента в сегмент со скоростью 1-5 Мбит/с.

Это не было большой проблемой: сети старались проектировать так, чтобы основной трафик сети проходил внутри сегментов (например, внутри одного сегмента размещали рабочие станции пользователей и сервера, с которыми эти пользователи преимущественно работали), а трафик между сегментами был не слишком значителен.

Однако постепенно ограничения производительности мостов становились узким местом сетей, вместе с тем сети все реже строили на базе физической шины (10Base5/10Base2) и все чаще на базе логической звезды (10BaseT), а применение логической звезды как раз и стимулирует использование многопортовых мостов в силу особенностей физической топологии.

Такие вот «многопортовые мосты», предназначенные в основном для применения в 10BaseT, были реализова-

ны аппаратно таким образом, чтобы работать на полной скорости провода и не быть узким местом сети. Именно такие устройства и называли коммутаторами (*англ. Switch*).

Таким образом коммутатор, это многопортовый мост с портами 10BaseT/10BaseF, который может передавать данные на скорости провода вследствие особенностей аппаратной реализации.

О том, как достигается возможность работать на скорости провода несмотря на значительную нагрузку, которую испытывают коммутаторы мы поговорим несколько позже, а пока давайте рассмотрим вопрос о том, как применялись мосты и коммутаторы в сетях в то время.

Когда основными физическими уровнями для построения локальных сетей были 10Base5/10Base2, мосты вполне очевидно применялись для того, чтобы разделить сеть на отдельные домены коллизий, позволяя улучшить время доступа к сети (за счет уменьшения конкуренции при доступе к среде), увеличить суммарную скорость обмена данными (так как внутри разных сегментов данные могут передаваться одновременно), уменьшить загруженность сети (по тем же причинам), а также для увеличения диаметра сети.

С переходом на 10BaseT, коммутаторы могли аналогично мостам из предыдущего примера применяться для соединения отдельных разделяемых сегментов, каждый из которых продолжал работать как независимый домен коллизий.

Но можно пойти дальше: что будет, если подключить к порту коммутатора не сегмент, а отдельную станцию, тогда такой сегмент, или домен коллизий, будет состо-

ять всего из двух портов: станции и коммутатора; такой подход называется микросегментация.

Следствием микросегментации станет уменьшение количества конкурирующих за среду передатчиков и увеличение общей производительности сети. Очевидно, что в таком сегменте резко уменьшится вероятность возникновения коллизий, однако совсем они не исчезнут.

Однако, раз уж наш сегмент стал таким маленьким и содержит РОВНО два устройства (порт коммутатора и порт рабочей станции), почему бы не задуматься о еще одной оптимизации? Как вы помните, в 10BaseT/F одна пара предназначена для передачи, а другая для приема, эта пара также слушает среду и, если слышит какой-либо сигнал, в момент своей передачи, то фиксирует коллизию. При микросегментации передающая пара адаптера узла непосредственно соединена с приемником коммутатора, прослушивающая — с его передатчиком.

А что, если вместо того, чтобы фиксировать коллизию в момент, когда навстречу нашей собственной передаче мы в приемной паре получаем сигнал, просто принимать этот сигнал и НЕ фиксировать коллизию? Это звучит очень необычно, ведь, если задуматься, мы только что предложили, не более и не менее как ОТКАЗАТЬСЯ от CSMA/CD!

Когда мы обсуждали коллизии в коаксиальном кабеле, все было очень просто: есть один кабель, если по нему передавать несколько сигналов с одним спектром они смешаются и их не получится разделить. При переходе на 10BaseT все уже не так очевидно: каждая станция соединена с концентратором ДВУМЯ парами, одна для



приема и одна для передачи. Однако и в этом случае мы все равно должны были фиксировать коллизию.

Действительно, пусть есть две станции А и В в классической сети 10BaseT, построенной с использованием концентратора. Пусть А передает данные, тогда этот сигнал есть в ПЕРЕДАЮЩЕЙ паре А и ПРИЕМНОЙ паре В. Пусть В тоже передает свой кадр, тогда он есть в ПЕРЕДАЮЩЕЙ паре В и ПРИЕМНОЙ паре А. На первый взгляд кажется, что сигналы нигде не смешиваются, но давайте добавим в сеть еще одну станцию С. В ее приемной паре есть кадр от А и там же есть кадр от В, таким образом станция С не может разделить эти сигналы, это — классическая коллизия, станции А и В должны ее распознать и обработать.

Но что, если наша сеть микросегментирована? А передает данные для С, кадр есть в ПЕРЕДАЮЩЕЙ паре А и приемной паре С. Пусть и В передает данные для С, кадр есть в ПЕРЕДАЮЩЕЙ паре В и НЕ ПОПАДЕТ в приемную пару С, так как эта пара уже занята передачей кадра от А к С, коммутатор просто буферизирует кадр от В к С и ждет, пока кадр от А будет передан С.

Пока все это вполне соответствует тому, о чем говорилось ранее и не дает нам ответа на вопрос, почему можно не фиксировать коллизии. Но давайте просто задумаемся: если ВСЕ порты сети имеют буферы и никогда не выдают ДВА сигнала в одну и ту же передающую пару (для кого-то она приемная), то ГДЕ в сети может возникнуть ситуация, когда в одной линии связи смешаются два сигнала?

Ведь коллизия это не просто прием одновременно с передачей, это ситауция, когда в одной линии связи

смешиваются без возможности разделения два сигнала с перекрывающимися спектрами. А если все порты в сети (и адаптеров и коммутаторов) имеют буферную память и никогда не выдают в один порт два кадра одновременно, а вместо этого коммутаторы будут буферизировать кадры, которые пока нельзя передать, то сигналы никогда не смешаются и, следовательно, коллизий больше не будет! А раз смещения сигналов в ситуации, когда я принимаю кадр одновременно с передачей больше нет, то зачем фиксировать коллизию, можно просто одновременно передавать и принимать данные!

Итак, достаточно просто отказаться от фиксации коллизий. Это очень далекоидущее решение, так как фактически отказ от фиксации коллизий убирает из Ethernet CSMA/CD и все недостатки, которые с ним связаны.

Во-первых, в идеальном случае суммарная пропускная способность между станцией и коммутатором удваивается, так как и станция и коммутатор могут одновременно передавать данные на скорости 10 Мбит/с. Во-вторых, станции или коммутатору больше не нужно дожидаться освобождения среды, получать к ней доступ по методу CSMA/CD, данные могут быть переданы по мере необходимости.

В-третьих теперь, когда коллизий нет, сеть нормально функционирует даже при нагрузке, приближающейся к 100% (в случае с коллизиями — не более 30–40%).

В-четвертых, общая производительность сети также вырастет, так как пропускная способность не будет теперь разделяться между всеми станциями сети, как это было в случае с концентраторами.

В-пятых, так как правило о PDV больше действует, это открывает возможности создания очень длинных линков Ethernet, ведь теперь ограничения на длину линка существуют только на физическом уровне, поэтому, при использовании сложных и дорогих приемников/передатчиков можно строить Ethernet линки длиной десятки километров, что важно для организации магистральных линий связи, ранее такие линии связи можно было использовать только с WAN технологиями, отказ от метода доступа открыл Ethernet возможность предлагать решения там, где еще недавно можно были применять лишь технологии глобальных сетей.

Фактически, отказавшись от метода доступа, Ethernet одним махом лишился всех своих очевидных недостатков (на самом деле не всех, например, Ethernet не позволяет приоритезировать важный трафик, гарантировать полосу пропускания для приложения, гарантировать задержки для приложения, но об этом мы пока говорить не будем).

Режим работы, при котором коллизии не фиксируются, а передавать данные можно в любой момент времени, называется полнодуплексный (full duplex), в противоположность полудуплексному режиму (half duplex), который использовался с CSMA/CD.

Итак, для того, чтобы воспользоваться преимуществами полнодуплексного режима, необходимо, во-первых, использовать для построения сети ТОЛЬКО коммутаторы, а во-вторых, перестроить работу MAC подуровня — отменить фиксацию и отработку коллизий.

Теперь давайте подумаем, что необходимо для реализации такого режима, и какие проблемы могут возникать при его использовании.

Работать в режиме полного дуплекса можно только в ситуации, когда к порту коммутатора подключена только одна станция. При подключении целого сегмента будет вновь возникать проблема разделения среды.

Оба порта, участвующих в обмене данными, должны поддерживать полнодуплексный режим, в противном случае, один из портов будет передавать данные, когда это ему будет необходимо, а второй, не поддерживающий этот режим, постоянно будет фиксировать коллизию.

Давайте рассмотрим еще один важный вопрос: откуда сетевой адаптер и порт коммутатора узнают, могут ли они работать в режиме полного дуплекса или должны выполнять CSMA/CD, ведь неверное принятое решение может серьезно навредить работоспособности линка да и сети в целом.

С появлением полного дуплекса в Ethernet появился механизм, называемый автопереговоры. Интерфейс, который поддерживает дуплекс с помощью специального сигнала сообщает об этом соседу, если сосед готов интерпретировать этот сигнал, значит он тоже поддерживает дуплекс и может быть установлен режим полного дуплекса, если же вторая сторона дуплекс не поддерживает, то первый получит от соседа вместо специального сигнала автопереговоров только Link Test и перейдет в полудуплексный режим работы. Обратите внимание, что автопереговоры проводятся не с помощью кадров Ethernet, а с помощью сигналов, спектр которых не пересекается со спектром передачи данных манчестерским кодом.

Давайте поговорим об одной проблеме, которую создает использование коммутаторов. Так как в полноду-

плексном режиме узлу разрешается ВСЕГДА отправлять кадры в коммутатор, то коммутаторы сети могут в этом режиме сталкиваться с перегрузками, не имея при этом никаких средств регулирования («притормаживания») потока кадров.

Причина перегрузок кроется в ограниченной пропускной способности отдельного порта. Легко может произойти ситуация, когда в какой-либо выходной порт коммутатора будет направляться трафик от многих портов с суммарной интенсивностью большей, чем 10 Мбит/с. Если одновременно несколько портов коммутатора принимают данные, которые нужно передать в один выходной порт, то буфер выходного порта коммутатора начнет быстро заполняться кадрами, ожидающими отправки.

А поскольку любой буфер рано или поздно заканчивается, то после его заполнения кадры просто начинают отбрасываться. Данные, которые переносились в этих кадрах, конечно, не пропадут, через некоторое время они будут переданы повторно, однако это потребует вмешательство верхних уровней модели OSI и в конечном итоге приведет к падению производительности сети, увеличение объема буфера ситуацию не спасет, так как всего лишь отсрочит неминуемое его переполнение.

Итак, налицо проблема, которой не могло существовать в классической сети Ethernet — проблема управления потоком данных. В классическом, CSMA/CD Ethernet потоком данных управлял сам метод доступа, не позволяя станциям передавать больше данных, чем сеть может передать. Теперь же несколько рабочих станций могут легко сгенерировать трафик, направленный на один единственный сервер и

пропускной способности линка сервера может не хватить, очевидно эту проблему нельзя оставлять без решения.

При разработке мостов для локальных сетей важной задачей было сохранение логики конечных узлов в неизменном виде, что позволяло узлам начать работать в сети с мостами ничего о них не зная. Поэтому у коммутаторов должен быть механизм, который позволял бы коммутатору «попросить» станции уменьшить интенсивность передачи данных если коммутатор (а точнее, выходной интерфейс) не справляется с потоком данных. Для начала рассмотрим более простой случай, пусть в центре некоторой сети применяется коммутатор, а к нему подключены концентраторы, к которым уже подключаются рабочие станции.

Ясно, что в этой схеме нет микросегментации и, как следствие, нет режима полного дуплекса, все достаточно интеллектуальные участники такой сети (узлы и коммутатор) должны фиксировать коллизии и выполнять CSMA/CD. Положим, узлы из нескольких сегментов на концентраторах отправляют данные серверу, который расположен в еще одном сегменте. Ясно, что в каждом сегменте только одна станция может передавать данные, но каждый сегмент на концентраторе может породить 10 Мбит/с, и, следовательно, этот поток данных не может быть передан в сегмент, в котором расположен сервер. С одной стороны коммутатор должен скомандовать устройствам за несколькими своими портами, чтобы они приостановили передачу данных, с другой стороны никакого специального инструментария для этого в Ethernet нет. Что же делать коммутатору?

Хотя в Ethernet нет специальных инструментов для этого, зато у коммутатора появляется возможность воз-

действовать на конечный узел с помощью механизмов методов доступа к среде, который конечный узел обязан обрабатывать. Эти приемы основаны на том, что конечные узлы строго соблюдают все параметры метода доступа к среде, а порты коммутатора могут по мере необходимости отступать от них.

Обычно применяются два основных способа управления потоком кадров:

- метод обратного давления на конечный узел (backpressure);
- агрессивный захват среды.

Метод обратного давления состоит в создании искусственных коллизий в сегменте, который чересчур интенсивно посылает кадры в коммутатор. Когда коммутатору необходимо «подавить» активность какого-либо порта, он искусственно генерирует коллизии на этот порт, посылая ему jam-последовательности.

Метод агрессивного захвата среды основан на «агрессивном» поведении порта коммутатора при захвате среды либо после окончания передачи очередного кадра, либо после коллизии. После успешной передачи кадра или коллизии, коммутатор вместо IPG в 9,6 мкс делает паузу, например, в 9,1 мкс и начинает передачу нового кадра. Компьютер не может захватить среду, так как выдерживает стандартный IPG в 9,6 мкс, после чего выясняется, что среда уже занята.

Таким образом, работая в полудуплексном режиме, коммутатор может управлять потоком данных от станций применяя хорошо знакомый нам метод доступа CSMA/CD с некоторыми вольностями.

Если же коммутатор работает в полнодуплексном режиме, то протокол работы конечных узлов уже все равно изменился, так как отключена фиксация коллизий, поэтому, для поддержки полнодуплексного режима работы коммутаторов имело бы смысл несколько модифицировать протокол взаимодействия узлов, встроив в него механизм управления потоком кадров.

В марте 1997 года принят стандарт IEEE 802.3x на управление потоком в полнодуплексных версиях протокола Ethernet. Он определяет два режима взаимодействия – синхронный и асинхронный.

В синхронном режиме для обеспечения управления потоком компоненты локальной сети обмениваются кадрами специального формата, которые называются кадры-паузы (pause frames). В поле DA такого кадра находится multicast адрес станций, которые поддерживают выполнение данной процедуры. Коммутатор или станция, отправляющая pause frame, предлагает «говорливой» станции значение времени, на которое она должна прекратить передачу.

Если станция не понимает таких кадров, используется асинхронный режим, предполагающий отправку с помощью кодов запрещенных кодов (например, 4B/5B) специальных команд «Прекратить передачу» и «Возобновить передачу». Сетевой адаптер или порт коммутатора, поддерживающий стандарт 802.3x и получивший команду «Приостановить передачу», должен прекратить передавать кадры вплоть до получения команды «Возобновить передачу».



Итак, на данный момент мы с вами ознакомились с возможными алгоритмами работы моста и коммутатора, теперь мы поговорим об ограничениях, которые возникают при их использовании.

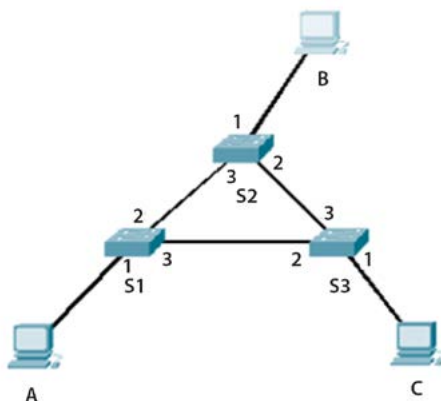
Многие сетевые технологии опираются от отправку широковещательных сообщений, т. е. кадров, которые должны достичь всех станций сети. Ясно, что коммутаторы должны обрабатывать такие кадры с помощью флудинга, т. е. отправлять такие кадры через все свои порты (кроме того, откуда кадр поступил, разумеется).

Однако такие сообщения занимают пропускную способность канала и нагружают все узлы в сети обработкой этих сообщений. И если станция в случае выхода из строя сетевого адаптера начинает постоянно генерировать широковещательный трафик, такая ситуация называется широковещательный шторм. Однако, если сеть очень велика, то даже небольшое количество широковещательного трафика, генерируемого каждой станцией может стать проблемой, если станций **ОЧЕНЬ** много, ведь **КАЖДОЕ** широковещательное сообщение достигает всех станций в сети независимо от того, коммутаторы в ней используются или концентраторы.

Единственным способом защиты от такой ситуации может быть административное установление максимально допустимого количества широковещательного трафика, теоретически это возможно, но практически весьма затруднительно определить какое количество этого трафика будет для сети безвредным. Ошибка же в определении этого количества может привести к некорректной работе всей составной сети.

Итак, первое ограничение, о котором нужно помнить, строя сеть на базе коммутаторов: хотя коммутаторы и делят сеть на домены коллизий, а типично вообще позволяют исключить понятие коллизий, мы все равно не можем строить гигантские сети с огромным количеством узлов с помощью коммутаторов. Тому есть много причин, но одна из них — коммутаторы не делят сеть на домены широковещательного трафика, каждая сеть на коммутаторах это один домен широковещания и при большом количестве станций порождаемый ими широковещательный «фон» может быть очень значителен. Для решения этой проблемы нам придется применять новые устройства, маршрутизаторы, но об этом мы будем говорить в одном из следующих курсов.

Второе ограничение при построении сетей на базе коммутаторов: невозможность поддержки петлеобразных связей между коммутаторами и, следовательно, обеспечения резервных связей в сети. Покажем это на примере небольшой сети из трех коммутаторов, которые соединены избыточными связями и подключенных к ним станций.



Предположим, что изначально станция А была выключена, станции В и С включены но кадрами не обменивались, а связки между коммутаторами S1 и S3 не было, и она по ошибке была добавлена непосредственно перед включением станции А. Посмотрим, что произойдет после того, как станция А будет включена.

Обычно, в процессе загрузки каждая станция отправляет несколько широковещательных кадров для определения некоторых параметров стека сетевых протоколов. После получения широковещательного кадра от станции А коммутатор S1 форвардит его, как и положено, через все свои порты, зафиксировав при этом в таблице коммутации, что станция с MAC адресом А находится за портом 1:

S1

Станция А	Порт 1
-----------	--------

Коммутатор S2 сделает тоже самое, его адресная таблица будет выглядеть следующим образом:

S2

Станция А	Порт 3
-----------	--------

и далее кадр отправится на коммутатор S3. Но S3 получает такой кадр и от S1 и от S2! Положим, сперва S3 получил этот кадр от S1, он отправляет его на все порты(!), теперь его таблица коммутации выглядит так:

S3

Станция А	Порт 2
-----------	--------

Но затем S3 получает копию этого же кадра от S2, S3 принимает новый кадр, также отправляет его на все порты (!), и изменяет в таблице коммутации выходной порт для станции А.

S3

Станция А	Порт 2
Станция А	Порт 3

Перед тем как принять это кадр, S3 отправил первый экземпляр этого кадра на S2, и S2 изменил свою таблицу коммутации:

S3

Станция А	Порт 3
Станция А	Порт 2

Таким образом, кадр возвращается к S1, который переносит станцию А теперь уже за третий порт, весь цикл повторяется заново и будет повторяться бесконечно.

S1

Станция А	Порт 1
Станция А	Порт 3

Время от времени широковещательные кадры отправляют также станции А и В, внося свой посильный вклад в ухудшение ситуации. Посылка в таких условиях unicast пакетов ТАКЖЕ приведет к их постоянной циркуляции в образовавшейся петле, это связано с тем, что адресные таблицы коммутаторов постоянно изменяются.

**Подведем итог.** Невозможно организовать отказоустойчивость в сети, построенной на коммутаторах, так как коммутаторы неспособны работать в сети с петлевидными связями. Для того, чтобы получить возможность строить сети с отказоустойчивыми связями на базе коммутаторов необходимы дополнительные механизмы, о которых речь пойдет позже.

Итак, на данный момент мы с вами познакомились с методами структуризации локальных сетей, принципами работы мостов и коммутаторов, их преимуществами и недостатками и разными режимами работы сети на основе коммутаторов. Теперь поговорим о том, какое применение нашли все эти новшества и обсудим дальнейшее развитие наиболее популярной технологии Ethernet.

## 3. История появления FastEthernet

---

В начале 90-х годов XX века встал вопрос о том, что пропускной способности сети Ethernet (10 Мбит/с) уже недостаточно для удовлетворения потребностей пользователей в высокоскоростной передаче данных.

Дело в том, что в этот период особенно интенсивными темпами стали развиваться компьютерные технологии в целом. Стали широко распространяться новые, более мощные компьютеры с новой, более скоростной шиной передачи данных, а чем быстрее компьютер обрабатывает данные «внутри», тем более быстрыми должны быть и внешние интерфейсы. И если для компьютеров в шине ISA скорости 10 Мбит/с для сетевого обмена хватало, то для компьютеров с шиной PCI, особенно с учетом появления новых сетевых приложений, этой пропускной способности уже было недостаточно.

И хотя применение мостов и коммутаторов позволило структурировать сети, уменьшить загруженность сегментов и, как следствие, несколько увеличить эффективную пропускную способность сетей в расчете на одну станцию, этого было явно недостаточно. В Ethernet вся сеть имеет пропускную способность 10 Мбит/с, которая делится между всеми станциями, а к тому времени даже одна станция без труда могла породить длительный трафик на скорости, превышающей 10 Мбит/с.

Необходимо было более радикальное решение, нежели структуризация сети, необходимо было увеличить битовую

скорость передачи данных в сетях. Ясно, что увеличивать скорость нужно было значительно, для того, чтобы новое решение не устарело за несколько лет.

Ясно, что новое решение может быть разработано «с нуля» или базироваться на уже существующем, возможно даже сохраняя некоторую обратную совместимость. За разработку «с нуля» говорило то, что Ethernet с его CSMA/CD не гарантирует станции доступа к среде за заданное наперед время, плохо переносит высокие нагрузки на сеть и накладывает серьезные ограничения на диаметр сети, за разработку преемственного решения говорила простота CSMA/CD, наличие на рынке инженеров, знакомых с этой технологией, наличие элементной базы (контроллеров), выполняющих логику MAC уровня Ethernet (которую можно использовать в преемственном решении), дешевизна данных решений и возможная обратная совместимость.

В результате наличия двух конкурирующих подходов к решению, возникло две группы разработчиков, что привело к появлению двух совершенно разных технологий, позволяющих передавать данные с высокой скоростью, а именно 100 Мбит/с. Эти две технологии отличаются степенью преемственности с классическим Ethernet.

Технология FastEthernet, предложенная такими компаниями как SynOptics, 3Com и некоторыми другими, входящими в созданное ими объединение FastEthernet Alliance, сохранила самую основу работы технологии Ethernet — метод доступа CSMA/CD, а технология 100VG-AnyLAN, разработанная группой специалистов под руководством компаний Hewlett-Packard и AT&T, отказалась от него, предложив взамен совершенно новый метод

получения доступа к среде — Demand Priority, позволяющий избавиться от некоторых недостатков CSMA/CD. В течении 1993 года комитет 802 института IEEE рассмотрел предложенные решения, и осенью 1995 года были приняты два стандарта - FastEthernet, сохранивший метод доступа CSMA/CD, получил статус дополнения к стандарту 802.3u, и совершенно новый стандарт 802.12, описывающий технологию 100VG-AnyLAN.

История сложилась таким образом, что технология FastEthernet получила широкое распространение, используется и по сей день (хотя и не часто, но ведь прошло уже более 20 лет!) и лежит в основе еще более быстрых сетевых технологий, которые применяются сегодня. Что же касается технологии 100VG-AnyLAN, то об этой технологии мы кратко поговорим позже, так как она интересная сегодня только с исторической точки зрения.

Что же изменилось в FastEthernet по сравнению с классическим Ethernet? Ясно, что изменился физический уровень, ведь изменилась скорость передачи битов, следовательно потенциально изменились линии связи и методы кодирования. Мы обязательно обсудим изменения в FastEthernet на физическом уровне, однако давайте начнем с канального уровня.

Как мы знаем, на канальном уровне Ethernet описан метод доступа, метод обработки коллизий и формат кадра. Все это осталось без изменений! FastEthernet использует тот же самый метод доступа к среде, тот же подход к распознаванию коллизий и тот же формат кадра, что и классический Ethernet, а это означает, что логика в FastEthernet осталась абсолютно той же, какой была в



классическом Ethernet. Не изменились и все таймеры, (разумеется, выраженные в битовых интервалах, а не в секундах), т. е. IPG по-прежнему составлял 96 bt и т. д.).

Итак, новая технология FastEthernet сохранила весь MAC-подуровень классического Ethernet, но пропускная способность была повышена до 100 Мбит/с. Однако, как мы знаем, метод доступа CSMA/CD накладывает ограничения на диаметр сети при заданном минимальном размере кадра данных.

Так как минимальный размер кадра данных фиксирован в Ethernet и не изменился в FastEthernet, то нам важно понять, как это отразилось на максимальной возможной длине сети FastEthernet.

Так как скорость передачи данных увеличилась в 10 раз, значит битовый интервал стал в 10 раз короче и теперь составляет 0.01 мкс, а время передачи кадра минимальной длины, которое в битовых интервалах, разумеется, не изменилось, в единицах времени стало тоже в 10 раз меньше, и теперь составляет не 57.5 мкс, а 5.75 секунд. Как мы знаем, для корректного распознавания коллизий должно выполняться правило  $T_{min} \geq PDV$  и  $T_{min}$  в этой формуле теперь в 10 раз меньше, нежели в Ethernet.

Однако, PDV, т. е. задержка, связанная с распространением сигналов в среде не зависит от скорости передачи данных, а определяется только скоростью распространения электромагнитных волн в линии связи, ясно, что эта величина никак не связана с битовой скоростью передачи данных, а определяется лишь фундаментальной физической константой и параметрами материала, из которого сделана линия связи.

Таким образом, так как  $T_{min}$  стал в 10 раз меньше, то для корректного распознавания коллизий и PDV в сети должен стать в 10 раз меньше, а это возможно только за счет уменьшения физических размеров сети, примерно в 10 раз, и если в Ethernet максимальный диаметр сети мог составлять 2500 метров, то в FastEthernet он никогда не может превышать 272 метра.

Конечно, с одной стороны это похоже на серьезный недостаток при проектировании крупных сетей. Тем не менее, на практике все обстоит не так плохо, как кажется. Дело в том, что в это же время очень широко стали распространяться локальные сети на основе коммутаторов. Использование коммутаторов сняло ограничения на общую длину сети, остались только ограничения на длину физических сегментов, соединяющих соседние устройства (сетевой адаптер — коммутатор или коммутатор — коммутатор). Поэтому при создании магистралей локальных сетей большой протяженности технология FastEthernet также активно применяли, но только совместно с коммутаторами.

Что же касается второго ограничения классического Ethernet, связанного с уменьшением межкадрового интервала, связанного с ресинхронизацией концентраторами, то здесь оно не актуально: в сети FastEthernet не может быть больше двух концентраторов из-за PDV (сеть стала короче из-за PDV, но длины сегментов по-прежнему не менее 100м, следовательно концентраторов можно использовать меньше), а два концентратора не могут внести заметный вклад в PVV. Однако, помимо этого все физические реализации FastEthernet синхронны, т. е. в паузах

между кадрами не тишина, а специальные запрещенные в потоке обычных данных символы, что дополнительно упрощает проблему сокращения межкадрового интервала.

Выше мы говорили о том, что появление коммутаторов добавило в Ethernet новый механизм — автопереговоры, позволяющий двум участникам линка выяснить автоматически, можно ли передавать данные поверх линка в полном дуплексе или в обычном режиме с CSMA/CD. Появление FastEthernet расширило возможности выбора режима работы с помощью автопереговоров: теперь стороны могут договариваться не только о режиме дуплекса, но и поддерживаемой скорости передачи данных, выбирая один из нескольких режимов:

- 10 Мбит/с half duplex;
- 10 Мбит/с full duplex;
- 100 Мбит/с half duplex;
- 100 Мбит/с full duplex.

Схема автопереговоров позволяет выбрать наиболее выгодный режим работы двум физически соединенным устройствам, которые поддерживают несколько стандартов физического уровня, отличающихся битовой скоростью и количеством витых пар. Устройство, инициирующее процесс автопереговоров, посылает серию специальных импульсов Fast Link Pulse burst (FLP), предлагая партнеру наиболее приоритетный с его точки зрения режим взаимодействия. В ответ партнер отправляет аналогичный FLP, подтверждая предложенный режим либо выдвигая свои условия. В конечном итоге будет выбран наиболее приоритетный из возможных режимов работы.

Однако, если узел поддерживает только технологию Ethernet 10Base-T, он не сможет отправить ответный FLP, так как не знает этой технологии. В этом случае станция-инициатор переговоров услышит в ответ только импульсы link test, переданные манчестерским кодом, и установит у себя соответствующий режим взаимодействия.

**Итак, подведем итог:** канальный уровень в FastEthernet практически не изменился по сравнению с таковым у Ethernet, все отличия, позволяющие увеличить скорость передачи данных сосредоточены на физическом уровне. Однако увеличение скорости сети, использующей CSMA/CD не могло не сказаться на уменьшении размера сети, что могло быть скомпенсировано применением коммутаторов.

Теперь рассмотрим физические реализации FastEthernet.

## 4. Физический уровень технологии FastEthernet

---

Как уже отмечалось, все отличия технологии FastEthernet от Ethernet были сосредоточены на физическом уровне. Уровни MAC и LLC в FastEthernet остались абсолютно теми же, и их описывают прежние главы стандартов 802.3 и 802.2. Но если уровни MAC и LLC в FastEthernet остались прежними, то реализации физического уровня полностью изменились.

Для начала технология FastEthernet больше не поддерживает физическую топологию «общая шина», используется только физическая топология «звезда», это означает, что коаксиальные кабели более не применяются, сеть строится только с помощью балансных или волоконно-оптических кабелей с применением концентраторов (или коммутаторов).

Существовал целый ряд физических реализации FastEthernet, некоторые реализации были описаны в стандарте IEEE 802, некоторые были приняты индустрией, но не описаны в документах IEEE. Реализации использовали одну из четырех возможных сред передачи данных:

- 100BaseFX, волоконно-оптический многомодовый кабель, используются два волокна;
- 100BaseSX, волоконно-оптический многомодовый кабель, используются два волокна, не является стандартом IEEE;

- 100BaseFB, волоконно-оптический одномодовый кабель, используется одно волокно, не является стандартом IEEE;
- 100BaseLX10, волоконно-оптический одномодовый кабель, используется одно волокно, не является стандартом IEEE;
- 100BaseTX, витая пара категории 5, используются две пары;
- 100BaseT4, витая пара категории 3, используются четыре пары;
- 100BaseT2, витая пара категории 3, используется две пары.

Некоторые из этих реализаций были весьма популярны, некоторые отчасти применяются и сегодня, некоторые имели очень ограниченную и специфическую область применения, сейчас мы кратко рассмотрим каждую реализацию.

### **100Base-FX**

Эта спецификация описывает работу FastEthernet в сети, построенной на основе многомодовой оптоволоконной линии связи.

Для приема и передачи данных используются два отдельных волокна, что позволяет сети работать как в полудуплексном, так и в полнодуплексном режиме. Используется лазер с длиной волны 1300 нм, что делает данное решение несовместимым с 10Base-FL.

Данные кодируются потенциальным кодом NRZI, будучи предварительно обработанными с помощью метода логического кодирования 4В/5В. Как вы помните, этот

логический код заменяет каждые 4 бита данных подуровня MAC на 5-битовые последовательности. Избыточный бит позволяет потом применить потенциальные коды для передачи сигналов без риска создать постоянный сигнал в линии связи.

Мы с вами уже говорили, что нельзя напрямую использовать «чистые» потенциальные коды для передачи данных из-за плохой самосинхронизации приемника и передатчика. При передаче длинной последовательности единиц или нулей в течение долгого времени сигнал не изменяется и приемник не может определить момент чтения очередного бита.

Применение избыточного кода решает проблему длительной последовательности нулей. При использовании пяти бит для кодирования шестнадцати исходных 4-битовых комбинаций, можно построить такую таблицу кодирования, в которой любой исходный 4-х битовый код представляется 5-ти битовым кодом с чередующимися нулями и единицами, тем самым обеспечивается синхронизация приемника с передатчиком.

Так как из 32 возможных комбинаций 5-битовых порций для кодирования порций исходных данных нужно только 16, то остальные 16 комбинаций в коде 4B/5B используются в служебных целях. Наличие служебных символов позволило использовать в спецификациях FastEthernet схему непрерывного обмена сигналами между передатчиком и приемником и при свободном состоянии среды.

В реализации 100BaseFX максимальная длина сегмента может составить 412 метров в режиме half duplex (это связано с распознаванием коллизий и правилом PDV, трансиверы вносят задержку в 100 битовых интер-

валов, каждый метр кабеля — 1 битовый интервал, так что суммарная задержка на таком линке может составить максимально разрешенные 512 bt). Однако, если коллизии регистрировать не нужно (в режиме полного дуплекса), то передавать данные можно на расстояние до 2 км (это уже ограничение физического уровня, связанные с затуханием сигнала в линии связи).

### **100Base-SX**

Данная реализация также использует два многомодовых волокна, но вместо лазера с длиной волны 1300 нм использовались светодиоды с длиной волны 850 нм, это приводило к тому, что данное решение было более дешевым, совместимым с 10Base-FL, но с помощью данного физического уровня можно было передавать данные на расстояния, не превышающие 550 м (разумеется, если речь идет о применении коммутаторов, при использовании концентраторов ограничение на длину сети снова равно 412 м, так как это ограничение связано с распознаванием коллизий и расчетом PDV).

Данная реализация не являлась стандартом IEEE и была предложена индустрией как эффективная и недорогая замена 10Base-FL.

### **100Base-BX**

Это очень специфическая реализация (не стандарт IEEE), которая использовала одно волокно одномодового кабеля для передачи данных в обоих направлениях. Два трансивера на концах такой линии связи не симметричны — один передает данные с помощью длины волны 1550



нм, второй — 1310 нм, тем самым удавалось достичь передачи в обоих направлениях с помощью единственного волокна.

Физические ограничения позволяли передавать данные на расстояние до 40 км, разумеется, эта технология применялась в качестве линка между коммутаторами, поэтому ограничения, связанные с PDV здесь не актуальны.

### **100Base-LX10**

Это еще одна реализация для построения длинных FastEthernet линков между коммутаторами, она использовала два одномодовых волокна, лазер с длиной волны 1310 нм и позволяла передавать данные на расстояние до 10 км (разумеется, без применения метода доступа, который, как мы помним, ограничивает максимальную длину линка в 412 метров).

### **100Base-TX**

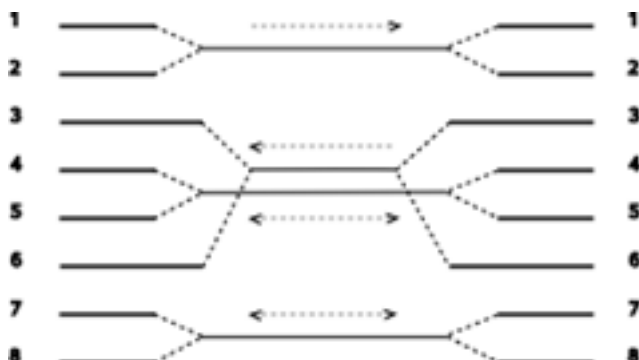
В качестве среды передачи данных эта спецификация использует неэкранированную или экранированную витую витую пару UTP cat. 5 с максимальной длиной сегмента 100 метров.

Используется физическое кодирование MLT-3, с предварительным логическим кодированием по методу 4В/5В, т. е. в линию передается 125 миллионов бит (из них 100 млн. полезных) с частотой первой гармоники, не превышающей  $125/4 = 32.25$  МГц, что хорошо укладывается в полосу пропускания витой пары 5-ой категории. Данная реализация совместима с 10Base-T и не так уже используется и по сей день.

## 100Base-T4

Спецификация 100Base-T4 появилась несколько позже всех других спецификаций FastEthernet. Основной целью ее разработчиков было обеспечение работы высокоскоростных технологий Ethernet поверх существующих кабельных систем 3-ей категории. Поскольку такая проводка была уже проложена в подавляющем числе зданий, именно на нее была ориентирована конкурирующая технология 100VG-AnyLAN. Поэтому спецификация 100Base-T4 была разработана в качестве альтернативной технологии, работающей поверх этой категории кабеля.

100Base-T4 использует метод кодирования 8В/6Т. Этот метод обладает достаточно узким спектром сигнала и позволяет достичь скорости до 33 Мбит/с при передаче по одной паре третьей категории. Заметьте, что если мы сможем передать по одной паре 33 Мбит/с в секунду, то для достижения скорости в 100 Мбит/с необходимо использовать три пары. Поэтому спецификация 100Base-T4 отличается несколько нестандартным использованием пар в кабеле. Передача данных происходит по всем четырем парам кабеля, при этом одна пара всегда передает сигнал, другая всегда принимает сигнал, и еще две используются в зависимости от направления потока данных и как передающие и как принимающие. Рассмотрим следующую схему:



На этом рисунке показано соединение сетевого адаптера станции 100Base-T4 с коммутатором (концентратором). Стрелки показывают направление потока данных по паре. В процессе передачи данных станцией сигнал будет передаваться парами 1–2 и так называемыми реверсными парами 4–5 и 7–8. Функция пары 3–6 состоит в прослушивании коллизии. Если же станция принимает данные, сигнал будет проходить по принимающей паре 3–6 и опять-таки по реверсным парам 4–5 и 7–8. Таким образом, скорость передачи данных по каждой из трех передающих пар приблизительно равна 33,3 Мбит/с, поэтому общая скорость протокола 100Base-T4 составляет 100 Мбит/с.

### 100Base-T2

Данная технология предназначена для передачи данных со скоростью 100 Мбит/с по двум витым парам категории 3, т. е. по линии связи, которая использовалась для передачи данных в технологии 10BaseT. Чаще всего кабельные системы, поверх которых работал 10BaseT содержали 4

витые пары категории 3, из которых 10BaseT использовал только 2 пары, поверх таких кабельных систем можно было передавать FastEthernet с помощью 100BaseT4. А вот для случая, когда существующая кабельная система имела только 2 витых пары категории 3 был разработан физический уровень 100BaseT2. Каждые 4 бита передаются за два такта с помощью метода физического кодирования PAM-5, что приводит к частоте первой гармоники, которая в четыре раза ниже битовой скорости передачи данных, одновременная передача по обоим парам позволяла достигать первой гармоники не выше 12.5 МГц при передаче данных со скоростью 50 Мбит/с по каждой из пар, при бодовой скорости 25 Мбод. Данная технология не получила массового распространения, как и 100BaseT4.

### **Подведем итоги по рассмотренным физическим реализациям FastEthernet.**

Оптические решения FastEthernet сегодня все еще применяются, например для подключения удаленного абонента в сети оператора связи. Если абонент хочет подключиться с помощью волоконно-оптического кабеля, то нередко применяется 100Base-FX линк, длина которого может, как мы говорили, достигать 2 км в рамках стандарта, кроме того на рынке присутствуют и более «дальние» трансиверы, позволяющие передавать данные и на большие расстояния (разумеется, с обоих концов такого удлиненного линка должны использоваться соответствующие нестандартные трансиверы), также очевидно, что такие линки возможны только при применении коммутаторов, так как распознавание коллизий в FastEthernet сетях такого

размера невозможно. Что касается остальных оптических FastEthernet, то все они были разработаны для решения специфических задач: 100Base-SX для применения вместо 10Base-FL (это уже давно не актуально), 100BaseBX/LX10 — для построения длинных магистральных линков, для таких линков пропускной способности FastEthernet уже давно не достаточно, поэтому FastEthernet здесь заменен на более скоростные реализации, о которых пойдет речь ниже.

Медные решения на базе FastEthernet также все еще применяются, конечно же речь идет о 100BaseTX, которую все еще используют в относительно старых компьютерах, коммутаторах и домашних маршрутизаторах. И хотя сегодня, даже для подключения обычного пользователя к сети все чаще применяют более высокоскоростные линки, 100BaseTX все еще остается нередко используемой технологией. Разумеется, 100BaseT4/T2, которые были разработаны для передачи данных на высокой скорости поверх очень старых линий связи третьей категории сегодня абсолютно не актуальны.

## 5. Методика расчета конфигурации сети FastEthernet

---

Теперь давайте поговорим о том, каким требованиям должна отвечать сеть FastEthernet для того, чтобы быть корректной. Для начала положим, что сеть строится на концентраторах, т. е. распознавать коллизии необходимо.

Очевидно, что при применении метода доступа CSMA/CD сеть должна быть, как и ранее, недогружена, максимальная длина кабеля в сегменте не должна превышать правил соответствующего физического уровня (для корректного распознавания сигнала), время двойного оборота сигнала (Path Delay Value, PDV) между двумя самыми удаленными друг от друга станциями сети не должно превышать длины минимального кадра и сокращение межкадрового интервала IPG (Path Variability Value, PVV) при прохождении последовательности кадров через все повторители должно быть не больше, чем 49 битовых интервалов. Последнее правило (о PVV) можно более не принимать во внимание — оно всегда выполняется: из-за увеличения скорости сеть стала «короче», более двух повторителей в сети FastEthernet вообще не может применяться, поэтому учет сокращения межкадрового интервала более не имеет смысла.

Давайте обсудим, какой вклад в PDV вносят концентраторы. Спецификации FastEthernet, о которых мы выше говорили, используют разные методы кодирова-

ния: 100Base-FX — NRZI (4В/5В), 100Base-TX — MLT3 (4В/5В), 100Base-T4 — 8В/6Т. Заметьте, что 100Base-FX/ТХ и 100Base-T4 используют разное логическое кодирование, следовательно, можно задаться вопросом, а каким образом повторитель будет работать с разным кодированием? Очевидно, что он должен работать либо с каким-то одним логическим кодом, либо поддерживать оба.

Таким образом, можно сказать, что повторители FastEthernet делятся на два класса, отличающихся поддержкой разных методов логического кодирования, а, соответственно, и типов физического уровня. Повторители I класса способны выполнять трансляцию логических кодов и работать со всеми спецификациями физического уровня FastEthernet, а повторители II класса могут поддерживать только какой-то один логический код — либо 8В/6Т, тогда доступны только порты 100Base-T4, либо 4В/5В, в этом случае можно использовать сегменты 100Base-FX/ТХ.

Однако процесс трансляции логических кодов занимает определенное время, поэтому повторители I класса будут вносить повышенные задержки (приблизительно 70 bt). Это лимитирует их использование, стандартом допускается использование только одного такого повторителя в одном домене коллизий.

Задержка, вносимая повторителями класса II значительно меньше, в зависимости от используемого типа физического уровня она варьируется от 33,5 bt до 46 bt. Поэтому разрешается устанавливать в один домен коллизий два повторителя этого класса, максимально допустимая длина для кабеля, их соединяющего, не должна превышать 5 метров.

Последнее условие на самом деле универсальное, справедливое для всех типов конфигураций, но если произвести необходимые расчеты, то можно показать, что для некоторых конфигураций эта длина может быть и больше. С другой стороны, если просто пользоваться именно этим ограничением, то вы никогда не ошибетесь. Для того, чтобы проводить какие либо расчеты нам нужны некоторые справочные данные для стандарта FastEthernet. Применяя наряду с повторителями коммутаторы или маршрутизаторы, можно было дополнительно разбивать сеть на домены коллизий, в каждом из которых устанавливать повторитель класса I или два повторителя класса II. Таким образом, удавалось избежать ограничений на общий диаметр сети.

Но опять, еще раз и еще раз напоминаю, для определения корректной конфигурации сети можно не руководствоваться правилами одного или двух хабов, а нужно рассчитывать время двойного оборота сети, PDV, как это было показано выше для сети Ethernet 10 Мбит/с.

Таблицы для расчета корректности сети FastEthernet приведены в приложении к уроку и не имеют на сегодняшний день практической ценности, так как современные сети не используют разделяемые сегменты и CSMA/CD, а используют только коммутаторы.



## 6. Технология GigabitEthernet (802.3Z)

---

Ясно, что через некоторое время и скорость 100Мб/с оказалась в некоторых случаях недостаточной. И это особенно ярко стало заметно при проектировании больших корпоративных сетей, со множеством серверов и рабочих станций. Магистралы таких сетей, работающие на скорости 100Мб/с, испытывали постоянные перегрузки из-за частых обращений клиентских станций, многие из которых также работали на такой скорости.

Возникла острая необходимость в разработке еще более скоростных технологий, и в середине 1996 года группа разработчиков во главе Cisco Systems и 3Com создала GigabitEthernet Alliance и принялась за работу.

В процессе работы над новой скоростной технологией появилось два отдельных подкомитета, один из них занимался вопросами передачи данных со скоростью 1000 Мб/с по оптоволоконным линиям связи (IEEE 802.3z), другой — реализацией такой скорости поверх витой пары UTP cat. 5 (802.3ab). Результатом работы этих подкомитетов стало принятие соответствующих стандартов: IEEE 802.3z (июнь 1998 г.) и 802.3ab (сентябрь 1999 г.). Помимо технологии GigabitEthernet также на рынке появилась технология 1000VG-AnyLAN, но никаких перспектив она не имела.

Задачей разработчиков вновь было максимальное сохранение преимуществ классического Ethernet при

десятикратном увеличении скорости. Ясно, что увеличение скорости передачи данных еще в 10 раз неизбежно потребует изменения физического уровня, однако можно ли по-прежнему не вносить изменений в реализацию канального уровня?

Вспомним, когда появился FastEthernet, это привело к примерно десятикратному уменьшению диаметра полудуплексной сети, ведь время передачи кадра минимальной длины уменьшилось в десять раз, поэтому необходимо было соответственно в десять раз уменьшить PDV для корректного распознавания коллизий, а это можно было сделать только уменьшив диаметр сети, с примерно 2500 метров до примерно 200 метров. Ясно, что если еще раз увеличить скорость передачи битов в десять раз, придется снова уменьшать диаметр сети примерно в 10 раз, но это, очевидно, неприемлемо!

Разработчики решили эту проблему следующим, очевидным, но неприятным способом — увеличением кадра минимальной длины, теперь кадр минимальной длины составлял не 64 байта, а 512 байт, таким образом PDV сети должен был менее, чем 4095 битовых интервала (длительность bt теперь равна 1 нс), это дало возможность получать сети диаметром около 200 метров при использовании одного повторителя, т. е. получить почти такой же результат, как и для FastEthernet.

Помимо этого изменения канального уровня, в GigabitEthernet появилось еще одно отличие от Ethernet/FastEthernet: узлам теперь разрешено захватывать среду для передачи не одного кадра, как ранее, теперь станция, получившая доступ к среде передачи данных может пере-

давать без IPG и освобождения среды несколько кадров подряд, общей длиной не более 8192 байт — предел, называемый Burst Length. В остальном канальный уровень GigabitEthernet не отличается от Ethernet/FastEthernet, все остальные отличия снова сосредоточены на физическом уровне.

Также нужно отметить, что в отличие от FastEthernet, который активно применялся в полудуплексном режиме на ранних этапах своего жизненного цикла, GigabitEthernet практически не применялся таким образом, основное применение GigabitEthernet связано с отказом от метода доступа и применением коммутаторов.

## 7. Физические реализации Gigabit Ethernet

---

Передача данных на столь высокой скорости была не простой задачей, особенно поверх витой пары 5 категории, поэтому первые физические реализации GigabitEthernet появились поверх волоконно-оптических кабелей (и еще одно, очень специфическое решение поверх витой пары очень малой длины для очень коротких линков). Физическая реализация поверх витой пары 5-ой категории при длине сегмента 100 метров появилась заметно позже оптических реализаций. Давайте кратко рассмотрим применявшиеся и применяемые сейчас физические реализации GigabitEthernet.

### **1000Base-CX**

Эта технология — первое решение для передачи данных со скоростью 1000 Мбит/с поверх медной экранированной витой пары, при этом максимальная длина линка могла составлять лишь 25 метров, ясно, что такой физический уровень подходил только для организации коротких линков, а не для построения полноценной сети. На сегодняшний день данная технология не применяется.

### **1000Base-SX**

Данный физический уровень предназначен для передачи данных поверх недорогих многомодовых волоконно-оптических линий связи на небольшие расстояния (220 метров для 62.5/125  $\mu\text{m}$  кабелей и 550 метров для

50/125  $\mu\text{m}$  кабелей, однако эти расстояния можно увеличить, выпуская более качественные и дорогие трансиверы) с использованием длин волн в диапазоне 770 нм — 860 нм. 1000Base-SX предназначена для организации оптических сетей внутри зданий, с помощью этой технологии можно производить оптические подключения рабочих станций и серверов к локальным сетям. Ясно, что построение сетей с большим диаметром возможно только с применением коммутаторов, впрочем, начиная с GigabitEthernet концентраторы практически не применялись.

### **1000Base-LX**

Данный физический уровень предназначен для передачи данных поверх более дорогих одномодовых кабелей на расстояния до 5 км с использованием лазеров с длиной волны 1270 нм — 1350 нм, ясно, что это решение предназначено для организации магистральных линий связи большой длины.

### **1000Base-LX10**

Данный физический уровень подобен 1000BaseLX, но может передавать данные на расстояние до 10 км. Прежде чем 1000Base-LX10 был принят как стандарт, долгое время он существовал на рынке под именем 1000BaseLX/LH или 1000Base-LH, поддерживаемый многими вендорами. Ясно, что область применения этой реализации аналогично предыдущему решению.

### **1000Base-EX**

Данная реализация физического уровня не является стандартом IEEE, однако используется многими вендорами

на рынке. Это решение очень похоже на 1000Base-LX10, однако позволяет передавать данные на расстояние до 40 км с использованием одномодового кабеля и лазеров с длиной волны 1310 нм.

### **1000Base-ZX**

Данная реализация физического уровня также не является стандартом IEEE, однако тоже используется многими вендорами. Это решение похоже на 1000Base-EX, но позволяет увеличить расстояние до 70 км с использованием одномодового кабеля и лазеров с длиной волны 1550 нм. Некоторые вендоры предлагают решения, позволяющие передавать данные на расстояние до 120 км, такую реализацию иногда называют 1000Base-EZX.

### **1000Base-T**

Этот физический уровень должен был обеспечить передачу данных со скоростью 1000 Мбит/с поверх существующих на тот момент линий связи категории 5. Эта задача оказалась крайне сложной, поэтому, во-первых, данный стандарт появился заметно позже первых оптических решений, а во-вторых, по большому счету, цель не была достигнута. Оказалось, что линии связи категории 5 не вполне пригодны для передачи данных на такой скорости, поэтому появился новый стандарт линий связи — категория 5е, поверх которых и работал 1000Base-T (более подробно о линиях связи категорий 5 и 5е, а также о том, чего не хватало линиям связи категории 5 для работы с 1000Base-T мы поговорим в следующем курсе). Не вдаваясь сейчас в детали отметим, что линии связи категории 5е, как и категории 5 имеют полосу

пропускания тоже равную 100 МГц, а для передачи 1000 Мбит/с по таким линиям связи приходится применять достаточно сложные методы. Для передачи данных в 1000Base-T используются все четыре пары, по каждой передается 250 Мбит/с. Даже это немалый поток данных для линии связи с полосой пропускания 100 МГц, данные передаются с помощью кода PAM-5, т. е. за одно изменение информационного параметра передается 2 бита, что порождает первую гармонику равную 62.5 МГц. Дуплекс достигается тем, что разрешена одновременная передача по всем парам в обе стороны, поэтому каждому адаптеру приходится вычитать передаваемый сигнал из принятого. Данный физический уровень очень широко распространен сегодня, практически каждый выпускаемый ноутбук/компьютер/домашний маршрутизатор оборудован именно такими интерфейсами. Данная технология поддерживает обратную совместимость с 10Base-T/100Base-TX и автопереговоры.

### **1000Base-TX**

Данный физический уровень был разработан TIA (Telecommunications Industry Association) как более эффективное решение для передачи данных поверх медных кабелей со скоростью 1000 Мбит/с. Данное решение использовало две однонаправленные пары (одна — прием, другая — передача), а не четыре двунаправленные, как 1000Base-T, при этом требовались линии связи категории 6. Однако это решение не имело коммерческого успеха, так как к моменту его появления на рынке, 1000Base-T завоевал заслуженную популярность.

# 10 GigabitEthernet и быстрее

---

Ясно, что технология GigabitEthernet и на сегодняшний день является хорошим решением для подключения рабочих станций и, часто, серверов, но для построения магистралей современных сетей, для соединений в рамках операторских сетей этой пропускной способности ясно недостаточно: сегодня даже домашние пользователи с легкостью могут получить от оператора связи подключение на скорости 100 Мбит/с, очевидно, операторам нужны более высокоскоростные технологии. В 2002 году было предложено решение 10GEthernet, которое, как ясно из названия, увеличивает скорость обмена данными еще в 10 раз.

Очевидно, что десятикратный прирост скорости относительно GigabitEthernet делает абсолютно невозможной даже теоретически работу в режиме half duplex. Действительно, даже с учетом увеличения кадра минимального размера, диаметр полудуплексной сети GigabitEthernet около 200 метров, следовательно диаметр полудуплексной сети 10G Ethernet не превысит примерно 20 метров. Даже если еще увеличить кадр минимального размера и довести его до максимально возможного значения 1518 байт, диаметр максимальный диаметр составил бы не более 60 метров. Поэтому решение было вполне очевидным: 10G Ethernet может работать ТОЛЬКО в режиме полного дуплекса. На практике уже GigabitEthernet применялся



только в full duplex, но для 10G Ethernet это стало единственной опцией. Если же учитывать отказ от CSMA/CD, то в остальном MAC подуровень не изменился, прежними остались формат кадра и все оставшиеся временные параметры, все прочие изменения сосредоточены на физическом уровне.

В 2002 году были определены следующие физические реализации 10G Ethernet:

- **10GBase-SR.**

Используется логическое кодирование 64В/66В, многомодовое волокно и лазер с длиной волны 850 нм. В зависимости от типа применяемого волокна допустимые расстояния 26 м, 33 м, 82 м, 300 м и 400 м. Данное решение предназначено, как видно, для работы на небольших расстояниях.

- **10GBase-LR.**

Используется логическое кодирование 64В/66В, одномодовое волокно и лазер с длиной волны 1310 нм. Максимальная длина линка — 10 км.

- **10GBase-ER.**

Используется логическое кодирование 64В/66В, одномодовое волокно и лазер с длиной волны 1550 нм. Максимальная длина линка — 40 км.

В 2004 году появилась первая реализация 10G поверх медного кабеля, но не балансного, а специального твинаксиального кабеля: 10GBase-CX4, максимальная длина линка не превышает 15 метров, область применения — короткие соединения.

Чуть позднее, в в 2006 году, появился стандарт физического уровня для передачи данных по витой паре категорий 6/6а: 10GBase-T с максимальными длинами линков соответственно 55 и 100 м, однако на практике это используется редко.

Есть и другие реализации физического уровня 10G Ethernet, предназначенные для решения различных задач: для ультракоротких соединений внутри одного устройства (10GBase-KX4, 10GBase-KR), для передачи данных поверх существующих WAN (10GBASE-SW, 10GBASE-LW, 10GBASE-EW), для создания очень длинных линков (10GBASE-ZR).

В 2010-2011 годах появились еще более быстрые физические уровни Ethernet: 40GBase- и 100GBase-. Ясно, что область применения таких технологий — соединения между операторами связи, очевидно, что говорить о CSMA/CD здесь не приходится, разумеется, поддерживается только full duplex. С точки зрения канального уровня ничего не изменилось, на данный момент определены следующие физические реализации:

- **40GBASE-CR4.** Используется 4 твинаксиальных кабеля, в каждом передается по 10 Гбит/с.
- **40GBASE-SR4.** Используется 4 многомодовых волокна, в каждом передается по 10 Гбит/с.
- **40GBASE-LR4.** Используется одно многомодовое волокно, в котором для передачи данных используется 4 независимых лазера на различных частотах.
- **40GBASE-ER4.** Используется одно многомодовое волокно, в котором для передачи данных используется 4 независимых лазера на различных частотах.

- **40GBASE-T.** Используется классическая витая пара (4 пары) категории 8.
- **100GBASE-CR4.** Используется 4 твинаксиальных кабеля, в каждом передается по 25 Гбит/с.
- **100GBASE-SR4.** Используется 4 многомодовых волокна, в каждом передается по 25 Гбит/с.
- **100GBASE-LR4.** Используется одно многомодовое волокно, в котором для передачи данных используется 4 независимых лазера на различных частотах, длина до 10 км.
- **100GBASE-ER4.** Аналогично предыдущему, но длина до 40 км.
- **100GBASE-CR10.** Используется 10 твинаксиальных кабелей, по каждому передается 10 Гбит/с.
- **100GBASE-SR10.** Используется 10 многомодовых кабелей, по каждому передается 10 Гбит/с.

Помимо указанных, есть и другие реализации физического уровня для 40G и 100G Ethernet, которые не являются стандартами IEEE, но приняты в индустрии. Важно понимать, что все эти решения применяются, в первую очередь, в операторских сетях.

**Подведем итоги:** сегодня мы познакомились с современными реализациями Ethernet. Мы познакомились с принципами работы коммутаторов и узнали, как Ethernet может работать вообще без метода доступа. Также изучили современные высокоскоростными реализациями Fast-, Gigabit-, 10G-, 40G- и 100G Ethernet.

## ПРИЛОЖЕНИЕ

Как и для технологии Ethernet 10 Мбит/с, комитет 802.3 дает исходные данные для расчета времени двойного оборота сигнала. Однако при этом сама форма представления этих данных и методика расчета несколько изменились. Комитет предоставляет данные об удвоенных задержках, вносимых каждым элементом сети, не разделяя сегменты сети на левый, правый и промежуточный. Кроме того, задержки, вносимые сетевыми адаптерами, учитывают преамбулы кадров, поэтому время двойного оборота при расчете конфигурации FastEthernet нужно сравнивать с величиной 512 битовых интервала (bt), то есть со временем передачи кадра минимальной длины без преамбулы.

Для повторителей класса I PDV можно рассчитать следующим образом:

- задержки, вносимые прохождением сигналов по кабелю, рассчитываются на основании данных таблицы, в которой учитывается удвоенное прохождение сигнала по кабелю:

Тип кабелей	Удвоенная задержка в bt на 1 м
UTP Cat. 3	1,14bt
UTP Cat. 4	1,14bt
UTP Cat. 5	1,112bt
Оптоволокно	1,112bt

- задержки, которые вносят два взаимодействующих через повторитель сетевых адаптера (или порта коммутатора), берутся из другой таблицы:

Тип сетевых адаптеров	Максимальная задержка при двойном обороте
Два адаптера TX/FX	100bt
Два адаптера T4	138 bt
Один адаптер TX/FX и один T4	127 bt

Учитывая, что удвоенная задержка, вносимая повторителем класса I, равна 140 bt, можно рассчитать время двойного оборота для произвольной конфигурации сети, естественно, учитывая максимально возможные длины непрерывных сегментов кабелей, приведенные в таблице. И если получившееся значение меньше 512, значит, по критерию распознавания коллизий сеть является корректной. Комитет 802.3 рекомендует оставлять еще запас в 4 bt для устойчиво работающей сети, но разрешает выбирать эту величину из диапазона от 0 до 5 bt.

Давайте, для примера рассчитаем рекомендуемую в таблице конфигурацию сети, состоящую из одного повторителя и двух оптоволоконных сегментов длиной по 136 метров:

- 1) каждый сегмент вносит задержку по 136 bt;
- 2) два сетевых адаптера дают совокупную задержку в 100 bt;
- 3) повторитель вносит задержку в 140 bt.

Итого, общая сумма задержек равна  $(2 \times 136) + 100 + 140 = 512$  bt, что говорит о том, что сеть корректна, но запас в этом случае принят равным 0.

Таким образом, мы научились с вами с помощью справочных данных рассчитывать корректную конфигурацию сетей FastEthernet. Однако необходимо учитывать тот факт, что все подобные вычисления проводятся в случае проектирования сети на концентраторах, а на практике такую сеть встретить практически невозможно. Поэтому все эти расчеты приводятся нами сугубо в образовательных целях.

Помните, что самым главным как для расчета конфигурации сетей классического Ethernet, так и сетей FastEthernet является определение выполнения критерия надежного распознавания коллизий. Все остальные правила и ограничения (1024 узла, 2500 м, 5-4-3, 5 метров между повторителями класса II, и т.п.) всего лишь помогают подобрать оптимальную конфигурацию сети, не являясь строгими критериями.





Урок №4

## Коммутация в локальных сетях. Базовый

© Компьютерная Академия «Шаг»

[www.itstep.org](http://www.itstep.org)

Все права на охраняемые авторским правом фото-, аудио- и видеопроизведения, фрагменты которых использованы в материале, принадлежат их законным владельцам. Фрагменты произведений используются в иллюстративных целях в объеме, оправданном поставленной задачей, в рамках учебного процесса и в учебных целях, в соответствии со ст. 1274 ч. 4 ГК РФ и ст. 21 и 23 Закона Украины «Про авторське право і суміжні права». Объем и способ цитируемых произведений соответствует принятым нормам, не наносит ущерба нормальному использованию объектов авторского права и не ущемляет законные интересы автора и правообладателей. Цитируемые фрагменты произведений на момент использования не могут быть заменены альтернативными, не охраняемыми авторским правом аналогами, и как таковые соответствуют критериям добросовестного использования и честного использования.

Все права защищены. Полное или частичное копирование материалов запрещено. Согласование использования произведений или их фрагментов производится с авторами и правообладателями. Согласованное использование материалов возможно только при указании источника.

Ответственность за несанкционированное копирование и коммерческое использование материалов определяется действующим законодательством Украины.