Лекция 2. Принципы функционирования ЛВС: протоколы и адресация.

Протокол – это набор правил, в соответствии с которым компьютеры обмениваются информацией. Эти правила включают формат, время и последовательность передачи данных, способы контроля и коррекции ошибок. В соответствии с моделью OSI (Open System Interconnection) существует семь уровней протоколов:

1. Физический уровень

Побитовая передача сигналов в кабелях: типы кодирования и физические характеристики сигналов, скорость передачи сигналов и т.д.

2. Канальный уровень

Передача кадров данных между сетевыми картами компьютеров. В самом общем виде кадр данных – это группа битов, состоящая из заголовка кадра и поля данных. В заголовке указывается адрес отправителя, адрес получателя, контрольная сумма и т.п. Канальный уровень обеспечивает получение доступа к общей среде передачи данных, обнаружение ошибок в кадрах данных, их повторную передачу и др. Канальный уровень – это аппаратное взаимодействие сетевая карта – сетевая карта.

3. Сетевой уровень

Сетевая ***логическая*** адресация сетевая карта – сетевая карта. Если на канальном уровне MAC-адрес сетевой карты физически "зашит" в ней производителем и не может изменяться, то на сетевом уровне сетевой карте компьютера может быть назначен любой логический адрес. При замене сетевой карты, MAC-адрес новой карты неизбежно будет другим, однако логический адрес новой карты можно оставить прежним, не нарушая адресацию в сети. Сетевой уровень также позволяет использовать в одной сети сегменты, построенные на различных протоколах канального уровня (например, объединить в единую сеть сегмент на сетевых картах Ethernet и сегмент на сетевых картах Token Ring). Кроме того, сетевой уровень отвечает за маршрутизацию (доставку) пакетов данных вне зависимости от сложности топологии сети.

4. Транспортный уровень.

Обеспечивает надежность доставки пакетов данных: установка виртуального канала передачи данных между сетевыми картами, контроль искажения или утери пакетов данных, повторная передача пакетов данных при необходимости.

5. Сеансовый уровень.

На практике используется редко (чаще всего сеансовый и представительский уровни объединяют с прикладным уровнем). Сеансовый уровень управляет диалогом сетевая карта – сетевая карта: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации, которые позволяют вставлять контрольные точки в длинные передачи данных, чтобы в случае сбоя можно было вернуться назад к последней контрольной точке, а не начинать все с начала.

6. Представительский уровень.

Позволяет менять форму представления информации, не меняя ее содержания. Например, преобразования кодировки ASCII в кодировку EBCDIC, или шифрование передаваемых по сети данных при помощи протокола SSL (Secure Socket Layer). При использовании SSL, с точки зрения прикладной программы ничего не меняется: взаимодействие между клиентом и сервером по сети происходит как обычно. Однако фактически, любые данные передаваемые программой в сеть, шифруются протоколом SSL на компьютере-отправителе, передаются по сети в зашифрованном виде, а затем дешифруются протоколом SSL на компьютере получателе, прозрачно (незаметно) для работающей сетевой программы.

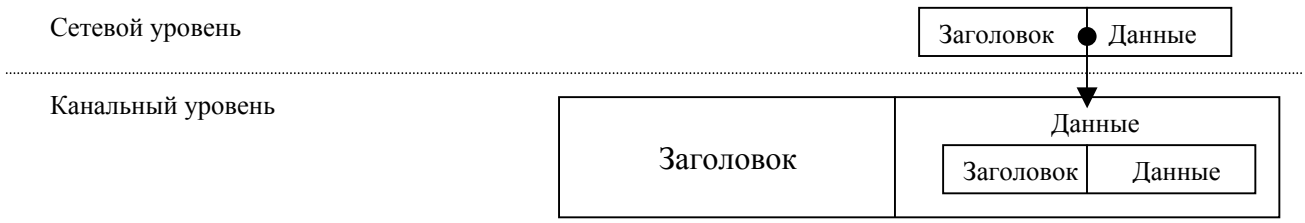
7. Прикладной уровень.

Набор разнообразных протоколов, при помощи которых взаимодействуют между собой прикладные программы. Каждая программа по желанию программиста может иметь свой собственный протокол или использовать один из широко-известных прикладных протоколов, например HTTP, SMTP, TELNET и др.

Модель OSI является международным стандартом, однако для практических целей, чаще всего пользуются упрощенной моделью в которой физический уровень подразумевается, но не рассматривается, а сеансовый и представительский уровни объединены с прикладным. Таким образом, упрощенная модель включает в себя:

- канальный уровень
- сетевой уровень
- транспортный уровень
- прикладной уровень

Важным понятием в многоуровневой модели протоколов является "инкапсуляция" пакетов. Чисто условно пакет можно представить в виде структуры [заголовок] – [данные]. В таком случае, инкапсуляцию можно представить следующей схемой:



При отправке, пакет сетевого уровня помещается в пакет (кадр) канального уровня, который и обеспечивает аппаратное взаимодействие сетевых карт, снимая эту задачу с протокола сетевого уровня. Протоколу сетевого уровня нет никакого дела до того, как реализован протокол канального уровня, а протокол канального уровня "не интересует" как работает протокол сетевого уровня – каждый выполняет свою часть работы. Инкапсуляция распространяется и на другие уровни: пакеты уровня приложения помещаются в пакеты транспортного уровня, которые в свою очередь помещаются в пакеты сетевого уровня, которые в свою очередь помещаются в пакеты канального уровня. Одним из следствий инкапсуляции является то, что при одном и том же протоколе канального уровня, может существовать несколько протоколов сетевого (транспортного, прикладного) уровня.

1. Протоколы канального уровня

1.1. Протокол Ethernet

Протокол Ethernet позволяет передавать данные со скоростью 10 Мбит/с и использовать следующие типы кабелей: толстый коаксиальный кабель (стандарт 10Base-5), тонкий коаксиал (стандарт 10Base-2), неэкранированную витую пару (стандарт 10Base-T), оптоволоконный кабель (стандарт 10Base-F).

Данные в протоколах канального уровня передаются в виде группы бит, организованных в кадр данных. Исторически существует 4 различных формата кадров Ethernet:

- кадр Ethernet DIX (Ethernet II) – один из первых форматов, стандарт фирм Digital, Intel и Xerox.
- кадр 802.3/LLC - международный стандарт.
- кадр Raw 802.3 (Novell 802.3) – стандарт фирмы Novell.
- кадр Ethernet SNAP – второй доработанный вариант международного стандарта.

Обычно сетевые карты автоматически распознают и поддерживают все четыре формата кадров. Для простоты изложения ограничимся рассмотрением самого простого по формату кадра Ethernet II, который имеет следующие поля:

Преамбула (для синхронизации) и признак начала кадра	Адрес назначения пакета	Адрес источника пакета	Тип пакета (указывает какому протоколу более высокого уровня принадлежит пакет)	Данные (передаваемая информация)	Контрольная сумма

Однако, помимо структуры кадра данных, в протоколе необходимо оговорить и порядок передачи этого кадра по сети. Основным принципом работы Ethernet является использование общей среды передачи данных разделяемой по времени, когда кадры данных передаются всеми компьютерами по общему кабелю. Особенно наглядно это проявляется при топологии "общая шина", хотя принцип сохраняется и при любой другой топологии. Впервые метод доступа к разделяемой общей среде был опробован во второй половине 60-х годов, в радиосети Aloha Гавайского университета, где общей средой передачи данных являлся радиоэфир. В 1975 году этот принцип был реализован и для коаксиального кабеля, в первой экспериментальной сети Ethernet Network фирмы Xerox.

В настоящее время сети Ethernet используется метод доступа CSMA/CD (Carrier Sense Multiply Access with Collision Detection) - коллективный доступ с проверкой несущей и обнаружением коллизий. Порядок передачи данных и коррекция ошибок происходит следующим образом: каждый кадр данных переданный в сеть получают все компьютеры, но только один из них распознает свой адрес и обрабатывает кадр. В каждый отдельный момент времени только один компьютер может передавать данные в сеть. Компьютер, который хочет передать кадр данных, прослушивает сеть и, если там отсутствует несущая частота (сигнал с частотой 5-10 Мгц), то он решает, что сеть свободна и начинает передавать кадр данных. Однако, может случится, что другой компьютер, не обнаружив несущей, тоже начнет передачу данных одновременно с первым. В таком случае, возникает столкновение (коллизия). Если один из передающих компьютеров обнаружил коллизию (передаваемый и наблюдаемый в кабеле сигнал отличаются), то он прекращает передачу кадра и усиливает ситуацию коллизии, посылкой в сеть специальных помех – последовательности из 32-бит (jam-последовательность), для того, чтобы и второй компьютер надежно обнаружил коллизию. После этого компьютеры ждут (каждый – случайное время) и повторяют передачу. Поскольку время – случайное (у каждого свое), то вероятность повторного столкновения невелика. Однако если столкновение произойдет снова (возможно с другими компьютерами), то следующий раз диапазон, в котором выбирается случайное время задержки, увеличится в 2 раза (после 10-й попытки увеличение не происходит, а после 16-й попытки кадр отбрасывается). В любом случае, время задержки, при возникновении коллизии невелико (максимум 52,4 миллисекунды) и незаметно для пользователя, однако при большой загрузке сети (начиная с 40 - 50%), слишком большая доля времени тратится на устранение коллизий и полезная пропускная способность падает. Более рациональным способом получения доступа к общей разделяемой среде является протокол Token Ring.

1.2. Протокол FastEthernet

Протокол Fast Ethernet был разработан совместными усилиями фирм SynOptics, 3Com (Fast Ethernet Alliance) и является развитием протокола Ethernet. Протокол Fast Ethernet позволяет передавать данные со скоростью 100 Мбит/с и использовать следующие типы кабелей: неэкранированную витую пару 5-й категории (стандарт 100Base-TX), неэкранированную витую пару 3-й категории (стандарт 100Base-T4), оптоволоконный кабель (стандарт 100Base-FX). Коаксиальный кабель в FastEthernet не поддерживается. Поддержка витой пары 3-й категории, несмотря на технические сложности, была реализована из-за того, что на западе, большинство уже проложенных телефонных кабелей, являются витой парой 3-й категории.

Метод доступа к разделяемой среде (CSMA/CD) в протоколе FastEternet остался прежним. Отличия от Ethernet заключаются в следующем:

- другой формат кадров
- другие временные параметры межкадрового и битового интервала (все параметры алгоритма доступа, измеренные в битовых интервалах сохранены прежними).
- признаком свободного состояния среды является передача по ней символа Idle (не занято), а не отсутствие сигнала, как в протоколе Ethernet.

Для совместимости со старыми сетевыми картами Ethernet, в протокол FastEthernet введена функция "автопереговоров" (auto-negotiation). При включении питания сетевой карты или по команде модуля управления сетевой карты начинается процесс "переговоров": сетевая карта посылает специальные служебные импульсы (FLP- fast link pulse burst), в которых предлагается самый приоритетный (с наибольшей скоростью передачи данных) протокол. Если второй компьютер поддерживает функцию "автопереговоров", то он ответит своими служебными импульсами, в которых согласится на предложенный протокол, или предложит другой (из поддерживаемых им). Если же на втором компьютере стоит старая сетевая карта Ethernet, не поддерживающая "автопереговоры", то ответа на запрос первого компьютера не последует, и он автоматически переключится на использование протокола Ethernet.

1.3. Протокол 100VG-AnyLan

Протокол 100VG-AnyLan был разработан совместными усилиями фирм Hewlett-Packard, AT&T и IBM.
И протокол FastEthernet и протокол 100VG-AnyLan являются развитием технологии Ethernet и позволяют работать на скорости 100 Мбит/с. Однако, если FastEthrnet ориентировался на минимальные изменения в протоколе Ethernet и совместимости со старыми сетевыми картами, то в протоколе 100VG-AnyLan, пользуясь сменой протоколов, была сделана попытка полностью отказаться от старых, и перейти к новым, более эффективным технологическим решениям.

Основным отличием 100VG-AnyLan является другой метод доступа к разделяемой среде - Demand Priority (приоритетный доступ по требованию), который обеспечивает более эффективное распределение пропускной способности сети, чем метод CSMA/CD. При доступе Demand Priority концентратору (hub-у) передаются функции арбитра, решающего проблему доступа к разделяемой среде. Сеть 100VG-AnyLAN состоит из центрального (корневого) концентратора, и соединенных с ним конечных узлов и других концентраторов (см. рис.). Допускаются три уровня каскадирования.

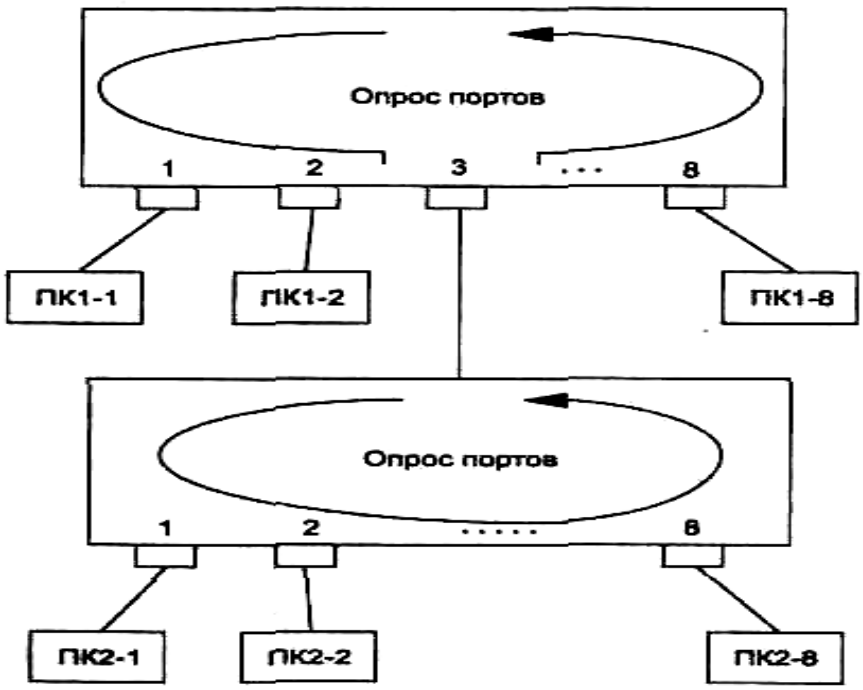


Рис.

Концентратор циклически выполняет опрос портов, к которым подключены компьютеры. Если к порту подключен другой концентратор, то опрос приостанавливается до завершения опроса концентратором нижнего уровня. Компьютер, желающий передать пакет, посылает специальный низкочастотный сигнал концентратору, запрашивая передачу кадра и указывая его приоритет: низкий (для обычных данных) или высокий (для данных, которые чувствительны к задержкам, например видеоизображение). Компьютер с низким уровнем приоритета, долго не имевший доступа к сети, получает высокий приоритет.

Если сеть свободна, то концентратор разрешает передачу пакета. Анализируется адрес назначения в пакете, и он передается на тот порт, к которому подключен соответствующий компьютер (адрес сетевой карты компьютера, подключенного к тому или иному порту, определяется автоматически, в момент физического подключения компьютера к концентратору). Если сеть занята, концентратор ставит полученный запрос в очередь. В очередь ставятся именно не сами кадры данных, а лишь запросы на их передачу. Запросы удовлетворяются в соответствии с порядком их поступления и с учетом приоритетов. У концентратора 100VG-AnyLan отсутствует внутренний буфера для хранения кадров, поэтому в каждый момент времени концентратор может принимать и передавать только один кадр данных – тот, до запроса на передачу которого дошла очередь (с учетом приоритетов).

В концентраторах 100VG-AnyLan поддерживаются кадры Ethernet и Token Ring (именно это обстоятельство дало добавку Any LAN в названии технологии). Каждый концентратор и сетевой адаптер 100VG-AnyLAN должен быть настроен либо на работу с кадрами Ethernet, либо с кадрами Token Ring, причем одновременно циркуляция обоих типов кадров не допускается. Другой особенностью является то, что кадры передаются не всем компьютерам сети, а только компьютеру назначения, что улучшает безопасность сети, т.к. кадры труднее перехватить при помощи анализаторов протоколов (снифферов).

Несмотря на много хороших технических решений, технология 100VG-AnyLAN не нашла большого количества сторонников и значительно уступает по популярности технологии Fast Ethernet.

1.4. Протокол GigabitEthernet

Протокол Gigabit Ethernet обеспечивает скорость передачи данных 1000 Мбит/с на всех основных типах кабельных систем: неэкранированная витая пара 5-ой категории, многомодовое и одномодовое оптоволокно (стандарты 1000Base-SX и 1000Base-LX), твинаксиальный кабель (коаксиальный кабель с двумя проводниками, каждый из которых помещен в экранирующую оплетку).

Протокол Gigabit Ethernet сохраняет максимально возможную преемственность с протоколами Ethernet и Fast Ethernet:

- сохраняются все форматы кадров Ethernet
- сохраняется метод доступа к разделяемой среде CSMA/CD. Поддерживается также полнодуплексный режим работы, когда данные передаются и принимаются одновременно (для отделения принимаемого сигнала от передаваемого сигнала, приемник вычитает из результирующего сигнала известный ему собственный сигнал).
- минимальный размер кадра увеличен (без учета преамбулы) с 64 до 512 байт. Для сокращения накладных расходов при использовании слишком длинных кадров для передачи небольших пакетов данных разработчики разрешили конечным узлам передавать несколько кадров подряд, без передачи среды другим станциям в режиме Burst Mode (монопольный пакетный режим). Если станции нужно передать несколько небольших пакетов данных, то она может не дополнять каждый кадр до размера в 512 байт (минимальный размер кадра), а передавать их подряд. Станция может передать подряд несколько кадров с общей длиной не более 65 536 бит или 8192 байт. Предел 8192 байт называется BurstLength. Если станция начала передавать кадр и предел BurstLength был достигнут в середине кадра, то кадр разрешается передать до конца.

1.5. Протокол Token Ring (High Speed Token Ring)

Использование протокола Token Ring позволяет карте работать на скоростях 4 и 16 Мбит/с, а протокола High Speed Token Ring – на скоростях 100 и 155 Мбит/с. Компания IBM является основным разработчиком протокола Token Ring, производя около 60 % сетевых адаптеров этой технологии.

Сеть Token Ring представляет собой кольцо: каждый компьютер соединен кабелем только с предыдущим и последующим компьютером в кольце. Физически это реализуется при помощи специальных концентраторов (см. рис.), которые обеспечивают целостность кольца даже при выключении или отказе одного из компьютеров, за счет обхода порта выключенного компьютера.

Принцип доступа к разделяемой среде – доступ с передачей маркера (token). Компьютер может начать передавать данные в сеть, только если получит от предыдущего компьютера в кольце "маркер" – специальный короткий пакет, свидетельствующий о том, что сеть свободна. Если компьютеру нечего передавать в сеть, то он передает маркер следующему компьютеру в кольце. Если компьютеру есть что передавать, то он уничтожает маркер и передает свой пакет в сеть. Пакет по битам ретранслируется по кольцу от компьютера к компьютеру, адресат получает пакет, устанавливает в пакете биты, подтверждающие, что пакет достиг адресата и передает пакет дальше по кольцу. Наконец, пакет возвращается к отправителю, который уничтожает его и передает в сеть новый маркер. Компьютер может и не передавать в сеть новый маркер, а продолжить передавать кадры данных до тех пор, пока не истечет время удержания

маркера (token holding time). После истечения времени удержания маркера компьютер обязан прекратить передачу собственных данных (текущий кадр разрешается завершить) и передать маркер далее по кольцу. Обычно время удержания маркера по умолчанию равно 10 мс.

В процессе работы сети, из-за сбоев, возможна потеря маркера. За наличие в сети маркера, причем единственной его копии, отвечает один из компьютеров - активный монитор. Если активный монитор не получает маркер в течение длительного времени (например 2,6 с), то он порождает новый маркер. Активный монитор выбирается во время инициализации кольца, как станция с максимальным значением MAC-адреса сетевой карты. Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора.

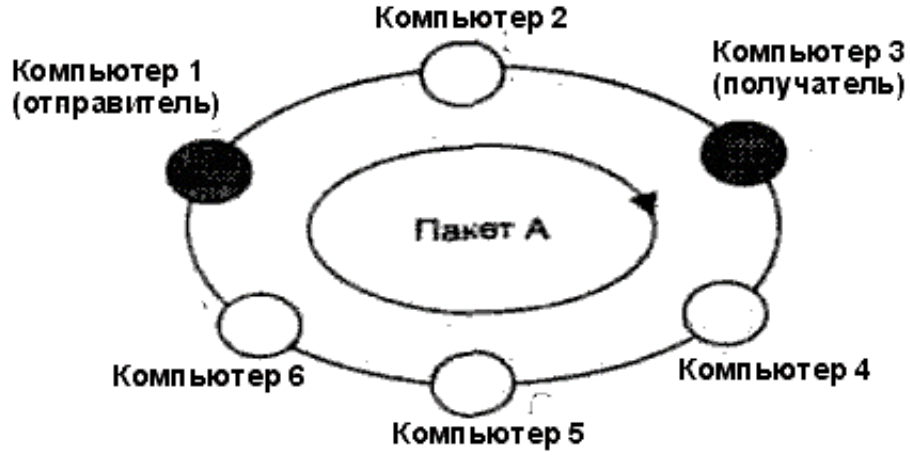


рис. Логическая структура сети Token Ring

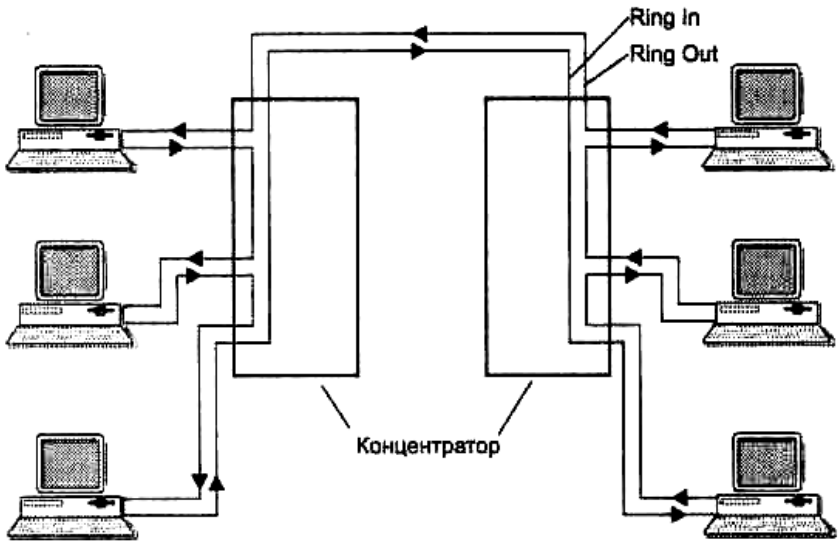


Рис. Физическая структура сети Token Ring

Описанный выше алгоритм доступа используется в сетях со скоростью 4 Мбит/с. В сетях со скорость 16 Мбит/с алгоритмы доступа более сложные: используется алгоритм доступа к кольцу, называемый алгоритмом раннего освобождения маркера (Early Token Release). Компьютер передает маркер доступа следующей станции сразу же после окончания передачи последнего бита кадра, не дожидаясь возвращения по кольцу этого кадра с битом подтверждения приема. В этом случае пропускная способность кольца используется более эффективно, так как по кольцу одновременно продвигаются кадры нескольких компьютеров. Тем не менее, свои кадры в каждый момент времени может генерировать только один компьютер — тот, который в данный момент владеет маркером доступа. Остальные компьютеры в это время только повторяют чужие кадры, так что принцип разделения кольца во времени сохраняется, ускоряется только процедура передачи владения кольцом.

Передаваемым кадрам, протокол верхнего уровня (например прикладного) может также назначить различные приоритеты: от 0 (низший) до 7 (высший). Маркер также всегда имеет некоторый уровень текущего приоритета и уровень резервного приоритета. При инициализации кольца основной и резервный приоритеты устанавливаются в ноль. Компьютер имеет право захватить переданный ему маркер только в том случае, если приоритет кадра, который он хочет передать, выше (или равен) текущему приоритету маркера. В противном случае компьютер обязан передать маркер следующему по кольцу компьютеру. Однако, даже если компьютер не захватил маркер, он может записать в поле резервного приоритета значение приоритета своего кадра (при условии, что предыдущие компьютеры не записали в это поле более высокий приоритет). При следующем обороте маркера резервный приоритет станет текущим и компьютер получит возможность захватить маркер.

Хотя механизм приоритетов в технологии Token Ring имеется, но он начинает работать только в том случае, когда приложение или прикладной протокол решают его использовать. Иначе все станции будут иметь равные права доступа к кольцу, что в основном и происходит на практике, так как большая часть приложений этим механизмом не пользуется.

Развитием протокола Token Ring стал протокол High-Speed Token Ring, который поддерживает скорости в 100 и 155 Мбит/с, сохраняя основные особенности технологии Token Ring 16 Мбит/с.

1.6. Протокол FDDI

Протокол FDDI (Fiber Distributed Data Interface) используется в оптоволоконных сетях и работает на скорости 100 Мбит/с. Исторически, когда скорости других протоколов ограничивались 10-16 Мбит/с, FDDI использовался на магистральных оптоволоконных сетях передачи данных.

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Сеть FDDI строится на основе ***двух*** оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец необходимо для повышения отказоустойчивости сети FDDI, и компьютеры, которые хотят воспользоваться этой повышенной надежностью могут (хотя это и не требуется) быть подключены к обоим кольцам.

В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного (Primary) кольца. Этот режим назван режимом Thru — «сквозным» или «транзитным». Вторичное кольцо (Secondary) в этом режиме не используется. В случае какого-либо отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ компьютера), первичное кольцо объединяется со вторичным (см. рис.), вновь образуя единое кольцо. Этот режим работы сети называется Wrap, то есть «свертывание» или «сворачивание» колец. Операция свертывания производится средствами концентраторов и/или сетевых карт FDDI. Для упрощения этой процедуры, данные по первичному кольцу всегда передаются в одном направлении, а по вторичному — в обратном (см. рис.). Поэтому при образовании общего кольца из двух колец, направление передачи данных по кольцам остается верным. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей.

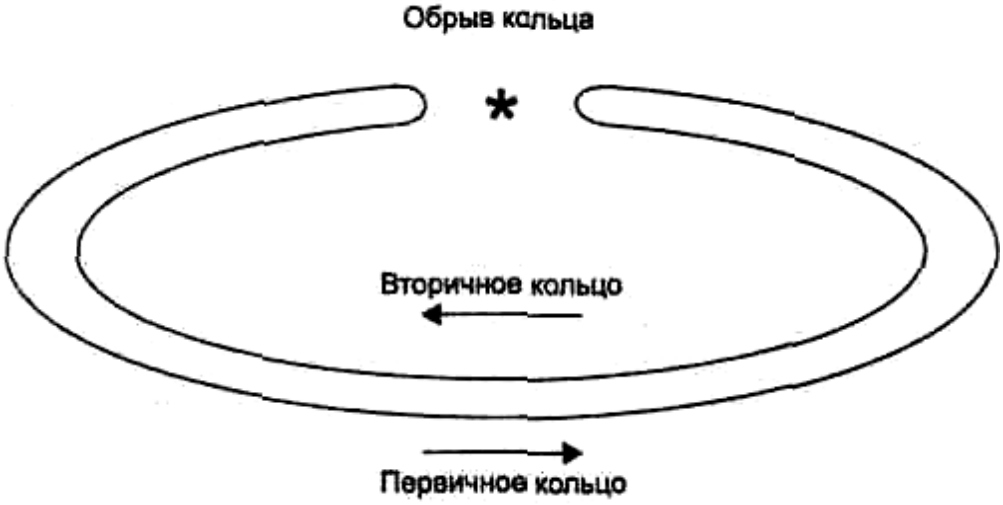


рис. Восстановление работоспособности сети FDDI при обрыве кольца.

Метод доступа к разделяемой среде в сети FDDI аналогичен методу доступа в сети Token Ring. Отличия заключаются в том, что время удержания маркера в сети FDDI не является постоянной величиной, как в

сети Token Ring, а зависит от загрузки кольца — при небольшой загрузке оно увеличивается, а при больших перегрузках может уменьшаться до нуля. В сети FDDI нет выделенного активного монитора — все компьютеры и концентраторы равноправны, и при обнаружении отклонений от нормы любой из них может начать процесс повторной инициализации сети, а затем и ее реконфигурации. В остальном пересылка кадров между станциями кольца полностью соответствует технологии Token Ring со скоростью 16 Мбит/с (применяется алгоритм раннего освобождения маркера).

На физическом уровне технология "сворачивания" колец реализуется специальными концентраторами. В стандарте FDDI допускаются два вида подсоединения компьютера к сети. Одновременное подключение к первичному и вторичному кольцам называется двойным подключением (Dual Attachment, DA). Компьютеры, подключенные таким образом, называются DAS (Dual Attachment Station), а концентраторы - DAC (Dual Attachment Concentrator). Подключение только к первичному кольцу называется одиночным подключением — Single Attachment, SA. Компьютеры, подключенные таким образом, называются SAS (Single Attachment Station), а концентраторы - SAC (Single Attachment Concentrator). Чтобы устройства легче было правильно присоединять к сети, их разъемы маркируются. Разъемы типа A и B должны быть у устройств с двойным подключением, разъем M (Master) имеется у концентратора для одиночного подключения станции, у которой ответный разъем должен иметь тип S (Slave). В случае однократного обрыва кабеля между устройствами с двойным подключением сеть FDDI сможет продолжить нормальную работу за счет автоматической реконфигурации внутренних путей передачи кадров между портами концентратора. При обрыве кабеля, идущего к компьютеру с одиночным подключением, он становится отрезанным от сети, а кольцо продолжает работать. Эта ситуация изображена на рисунке ниже.

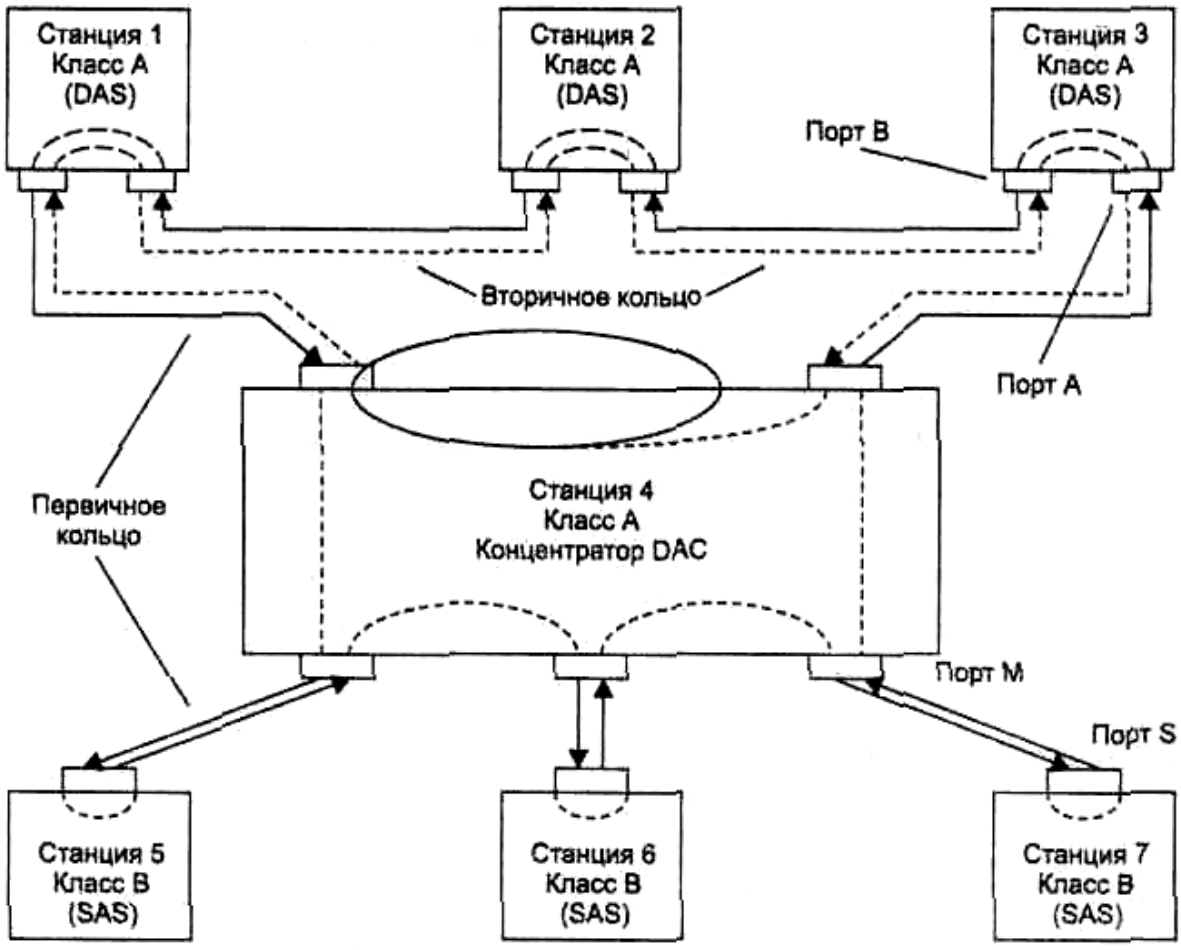


рис. Исходное подключение компьютеров к сети (до обрыва).

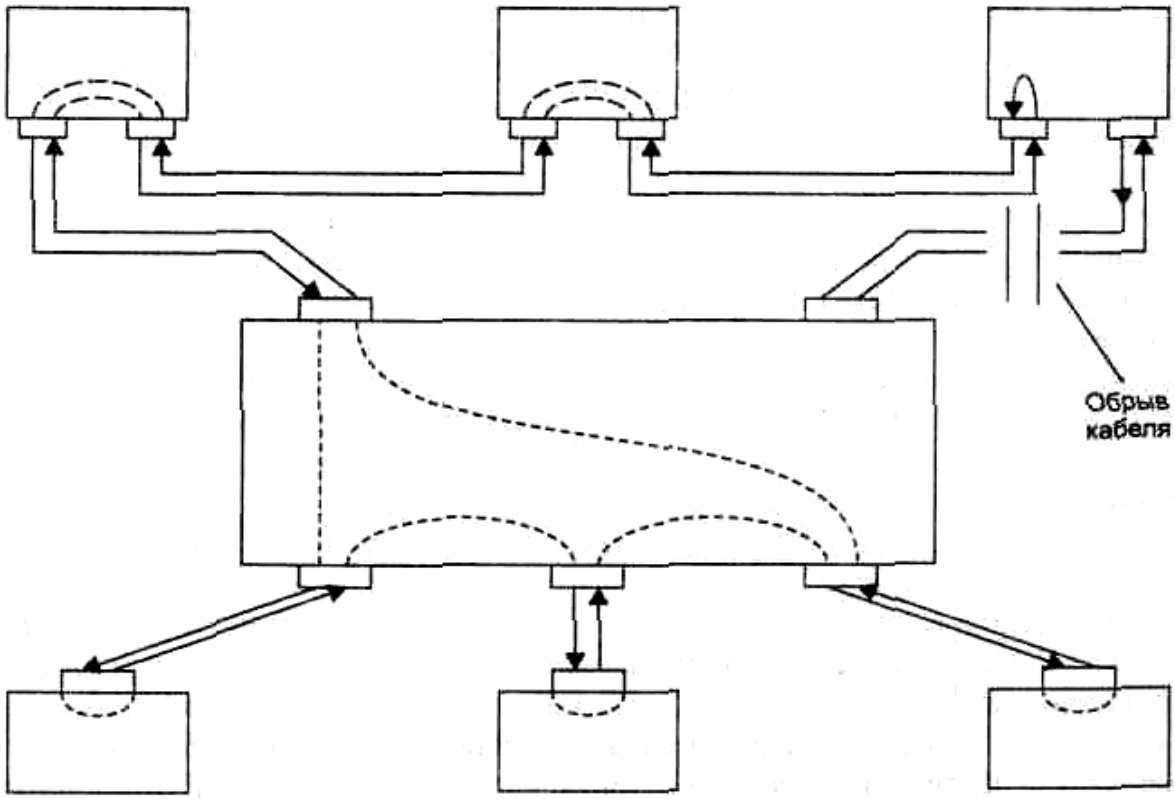


рис. Реконфигурация сети в случае обрыва.

1.7. Протоколы SLIP и PPP

Основное отличие протоколов SLIP и PPP от рассмотренных выше протоколов – это то, что они поддерживают связь "точка-точка", когда сетевой кабель используется для передачи информации только между двумя компьютерами (или другим сетевым оборудованием), соединенным этим кабелем. Такое соединение характерно при подключении к Internet по телефонной линии, при соединении локальных сетей между собой по выделенным или коммутируемым линиям, а также в сетях X.25, Frame Relay и ATM (см. далее в лекциях). Существует большое количество протоколов канального уровня для соединения "точка-точка", однако здесь мы ограничимся рассмотрением только SLIP и PPP.

SLIP (Serial Line IP) – протокол канального уровня, который позволяет использовать последовательную линию передачи данных (телефонную линию) для связи с другими компьютерами по протоколу IP (протокол сетевого уровня). SLIP появился достаточно давно, для связи между Unix – компьютерами по телефонным линиям и, в настоящее время, является устаревшим, т.к. не позволяет использовать протоколы сетевого уровня, отличные от IP, не позволяет согласовывать IP – адреса сторон и имеет слабую схему аутентификации (подтверждения личности) пользователя, заключающуюся в пересылке по сети имени и пароля пользователя. Таким образом, имя и пароль (даже зашифрованный) могут быть перехвачены и повторно использованы злоумышленником, или он может просто дождаться, пока пользователь пройдет аутентификацию, а затем отключить его и самому подключится от имени пользователя. Поэтому, большинство провайдеров Internet для подключения к своим машинам используют протокол PPP.

Протокол канального уровня PPP (Point to Point Protocol – протокол точка-точка) позволяет использовать не только протокол IP, но также и другие протоколы сетевого уровня (IPX, AppleTalk и др.). Достигается это за счет того, что в каждом кадре сообщения хранится не только 16-битная контрольная сумма, но и поле, задающее тип сетевого протокола. Протокол PPP также поддерживает сжатие заголовков IP-пакетов по методу Ван Джакобсона (VJ-сжатие), а также позволяет согласовать максимальный размер передаваемых дейтаграмм, IP-адреса сторон и др. Аутентификация в протоколе PPP является двусторонней, т.е. каждая из сторон может потребовать аутентификации другой. Процедура аутентификации проходит по одной из двух схем:

а) PAP (Password Authentication Protocol) – в начале соединения на сервер посылается имя пользователя и (возможно зашифрованный) пароль.

б) CHAP (Challenge Handshake Authentication Protocol) – в начале соединения сервер посылает клиенту случайный запрос (challenge). Клиент шифрует свой пароль, используя однонаправленную хэш-функцию (функция у которой по значению Y невозможно определить X) и запрос, в качестве ключа шифрования. Зашифрованный отклик (response) передается серверу, который, имея в своей базе данных пароль клиента, выполняет те же операции и, если полученный от клиента отклик совпадает с вычисленным сервером, то аутентификация считается успешной. Таким образом, пароль по линиям связи не передается. Даже если

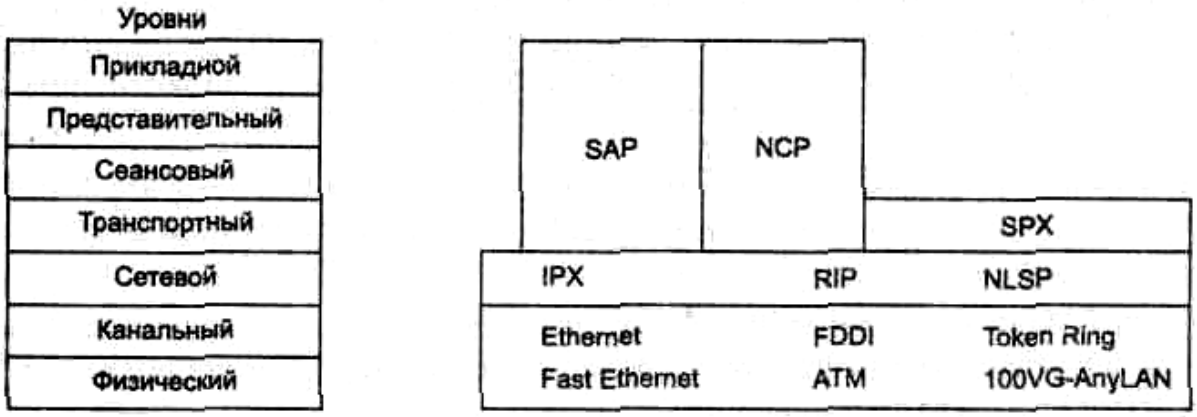
отклик клиента и будет перехвачен, то в следующий раз использовать его не удастся, т.к. запрос сервера будет другим. Определить же пароль на основании отклика – невозможно, т.к. хэш-функция шифрует данные только "в одну сторону". Для предотвращения вмешательства в соединение уже после прохождения клиентом аутентификации, в схеме CHAP сервер регулярно посылает испытательные запросы через равные промежутки времени. При отсутствии отклика или неверном отклике соединение прерывается.

2. Протоколы сетевого и транспортного уровня

Как и для канального уровня, существует несколько протоколов сетевого уровня. На практике, протокол сетевого уровня чаще всего разрабатывается и используется в паре с соответствующими протоколами транспортного, а иногда и прикладного уровня, образуя стек протоколов. В качестве примера можно привести следующие протоколы: IPX/SPX, NetBIOS/SMB, TCP/IP.

2.1. Стек протоколов IPX/SPX

Был разработан фирмой Novell для сетевой ОС NetWare, оптимизирован для использования в небольших локальных сетях, однако не удобен для глобальных сетей. Включает в себя протоколы IPX, SPX, SAP, NCP,



Протокол IPX (Internetwork Packet Exchange —межсетевой обмен пакетами) – протоколов сетевого уровня, поддерживает обмен пакетами (датаграммами) без установления канала связи и гарантии доставки пакета. Протокол IPX также отвечает за адресацию в сетях Net Ware. Адрес имеет формат: номер сети (задается администратором сети), адрес сетевой карты (определяется автоматически), номер сокета (идентифицирует приложение, пославшее пакет). Протокол IPX – аналог протокола IP из стека TCP/IP. Протокол IPX самый быстрый и экономит память, однако не дает гарантии доставки сообщения. За восстановлением утерянных или испорченных пакетов должен следить сам программист. Использование протокола SPX избавляет программиста от этой необходимости.

Протокол SPX (Sequenced Packet Exchange — последовательный обмен пакетами) – протокол транспортного уровня, поддерживает установление логического канала связи между компьютерами для обмена данными, коррекцию ошибок и, при необходимости, повторную передачу пакетов (аналог протокола TCP из стека TCP/IP).

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

Протоколы RIP (Routing Information Protocol) и NLSP (NetWare Link Service Protocol) отвечают за управление маршрутизацией (выбор маршрута доставки) пакетов, однако подробно здесь рассматриваться не будут. Протокол RIP реализован также и в стеке протоколов TCP/IP, а протокол NLSP аналогичен протоколу OSPF сетей TCP/IP.

2.2. Стек протоколов NetBIOS / SMB

Применяется фирмой Microsoft в своих сетевых ОС. В частности "сетевое окружение" работает при помощи этого протокола. NetBIOS включает в себя протоколы сетевого и транспортного уровня. Обеспечивает поддержку имен: каждая из рабочих станций в ЛВС может иметь одно или несколько имен (эти имена хранятся NetBIOS в таблице, в формате адрес сетевого адаптера – имя NetBIOS). Обеспечивает как обмен датаграммами, без установления канала связи и гарантии доставки сообщений, так и передачу пакетов с

установление логического канала связи между компьютерами с коррекцией ошибок и повторной передачей пакетов, при необходимости.

2.3. Стек протоколов TCP/IP

Протокол TCP/IP (Transmission Control Protocol/Internet Protocol – протокол контроля передачи данных / протокол передачи данных между сетями, Internet) разрабатывался Министерством Обороны США для глобальной сети ARPANET, и впоследствии стал основным протоколом, применяющимся в Internet. В состав стека протоколов TCP/IP входят протоколы: IP и ICMP – сетевой уровень, TCP и UDP – транспортный уровень. Ниже стек протоколов TCP/IP будет рассмотрено подробнее.

2.3.1. Протокол IP (ICMP)

Протокол IP отвечает за адресацию в сети и доставку пакетов между компьютерами сети, без установления соединения и гарантий доставки пакета. При использовании протокола IP, каждый компьютер в рамках сети должен иметь уникальный IP – адрес, представляющий собой 32-битное число. Для удобства чтения, IP адрес разбивают на четыре 8 битовых числа, называемых октетами, например 149.76.12.4. В локальной сети, которая не подключена к Internet или другим сетям, Вы можете назначать IP-адреса произвольно (главное, чтобы они не совпадали). Однако в Internet, IP-адреса выделяются централизовано, организацией InterNIC. InterNIC выдает адреса не на каждый отдельный компьютер, а в целом на локальную сеть.

В IP-адресе выделяют две части: сетевую часть (адрес локальной сети) и адрес компьютера в сети. Сетевая часть адреса может иметь переменную длину, которая зависит от класса IP-адреса и маски подсети. Выделяют следующие классы IP-адресов:

Класс A

включает сети с адресами от 1.0.0.0 до 127.0.0.0. Сетевой номер содержится в первом октете (1-127), что предусматривает 126 сетей по 1.6 миллионов компьютеров в каждой. Стандартная маска подсети для адреса класса имеет вид 255.0.0.0.

Класс B

Включает сети с адресами от 128.0.0.0 до 191.255.0.0. Сетевой номер находится в первых двух октетах (128.0 – 191.255), что предусматривает 16320 сетей с 65024 компьютерами в каждой. Стандартная маска подсети для адреса класса имеет вид 255.255.0.0.

Класс C

Включает сети с адресами от 192.0.0.0 до 223.255.255.0. Сетевой номер содержится в первых трех октетах (192.0.0 - 223.255.255). Это предполагает почти 2 миллиона сетей по 254 компьютеров в каждой. Стандартная маска подсети для адреса класса имеет вид 255.255.255.0.

Классы D

Включает адреса от 224.0.0.0 до 239.255.255.0. Эти адреса являются групповыми (multicast). Если нескольким компьютерам в сети назначен один и тот же групповой адрес, то пакет, адресованный на этот адрес, получат все компьютеры. Такие адреса в локальных сетях используются редко и зарезервированы для того времени, когда технические возможности сети Internet позволят организовывать теле- и радиовещание на группы компьютеров.

Классы E и F

Адреса попадающие в диапазон от 240.0.0.0 до 254.0.0.0 являются или экспериментальным, или сохранены для будущего использования и не определяют какую-либо сеть.

В примерах выше упоминалась "стандартная" маска подсети. Такая маска полностью соответствует классу адреса и может определяться автоматически, на основании анализа диапазона, в котором находится адрес. Казалось бы нет никакого смысла определять маску подсети вручную и вообще вводить такое понятие. Однако существуют ситуации, когда маска подсети будет отличаться от "стандартной". Допустим, у вас имеется сеть класса B (65024 компьютера) с IP-адресом 172.16.0.0 и вы хотите разбить ее на несколько подсетей, для разных филиалов предприятия. Стандартная маска подсети для адреса класса B равна 255.255.0.0 и адрес 172.16.1.0 интерпретируется, как компьютер с адресом 1.0 в сети с адресом 172.16. Однако если задать маску подсети равную 255.255.255.0, то этот IP-адрес прочитается как подсеть 172.16.1, содержащая 254 компьютера с адресами от 1 до 254. Таким образом, перед тем как решить является ли IP-адрес адресом конкретного компьютера или адресом сети, необходимо взглянуть на маску подсети, которая может отличаться от стандартной. Более того, маска подсети может не обязательно заканчиваться на границе байта. Маска всегда рассматривается в двоичном выражении, где единицы в октетах соответствуют полю адреса сети, а нули – полю адреса компьютера (см. табл.).

	Десятичное представление	Двоичное представление
IP-адрес	172 . 16 . 96 . 0	10101100 . 00010000 . 01100000 . 00000000
Маска подсети	255 . 255. 192 . 0	11111111 . 11111111 . 11000000 . 00000000
Интерпретация адреса: - адрес подсети - адрес компьютера	 172 . 16 . 1 32 . 0	 10101100 . 00010000 . 01 100000 . 00000000

Помимо адресов из классов A,B,C,D, E, F, ссуществует также несколько зарезервированных адресов. IP-адрес в котором все биты октеты адреса компьютера равны 0 относится ко всей сети, а где все биты октеты адреса компьютера равны 1 назван широковещательным (broadcast) адресом. Он относится к

каждому компьютеру сети. Таким образом, 149.76.255.255 - не существующий адрес компьютера, который относится ко всем компьютерам из сети 149.76.0.0.

Имеются еще два зарезервированных IP-адреса, 0.0.0.0 и 127.0.0.0. Первый назван путь пакетов по умолчанию (default route), второй - кольцевым (loopback) адресом или ссылкой на самого себя. В несуществующей сети 127.0.0.0, адрес 127.0.0.1 будет назначен специальному интерфейсу, который действует подобно закрытому кругообороту. Любой IP пакет переданный на этот адрес будет возвращен на этот же компьютер так, как если бы пакет пришел откуда-то из сети. Это позволяет тестировать сетевое программное обеспечение без использования "реальной" сети.

Также имеется ряд "серых" IP-адресов, которые зарезервированы для использования только в локальных сетях. Пакеты с "серыми" адресами не передаются маршрутизаторами Internet. К таким адресам относятся:

Сеть класса A 10.0.0.0
Сеть класса B от 172.16.0.0 до 172.31.0.0
Сеть класса C от 192.168.0.0 до 192.168.255.0

По соображениям безопасности, рекомендуется использовать в локальных сетях только "серые" адреса. В таком случае прямой доступ из Internet к компьютерам ЛВС, в обход прокси-сервера организации, будет невозможен. При доставке, пакет от компьютера злоумышленника к компьютеру жертвы пройдет не один маршрутизатор Internet (алгоритмы маршрутизации см. ниже). Если адрес компьютера жертвы "серый", то первый же маршрутизатор Internet заблокирует пакет и не станет передавать его дальше. Таким образом, злоумышленнику придется сначала соединиться с прокси-сервером организации (на котором установлены средства аутентификации (проверки личности) пользователя, межсетевой экран и т.п.), и только прокси-сервер сможет обеспечивать контролируемое и протоколируемое взаимодействие между компьютером ЛВС и Internet, благодаря технологии NAT. Network Adress Translation (NAT) – это подмена в отправляемых и принимаемых пакетах данных "серых" IP-адресов компьютеров локальной сети на "реальный" IP-адрес прокси-сервера в сети Internet (более подробно см. далее в лекциях). Использование "серых" адресов также гарантирует, что даже если сообщение от одного компьютера ЛВС, к другом компьютеру ЛВС случайно попадет в каналы связи с Internet, то оно не будет передано дальше и не будет получено другой машиной, со случайно совпадающим IP-адресом.

Кроме адресации компьютеров в сети, протокол IP также отвечает за маршрутизацию (выбор маршрута доставки) пакетов данных в сетях с произвольной топологией. Маршрутизация происходит на основании специальных таблиц маршрутизации либо программно (сетевой операционной системой), либо при помощи специальных сетевых устройств – маршрутизаторов (подробнее маршрутизаторы будут рассмотрены далее в лекциях). Рассмотрим, как происходит доставка пакета по протоколу IP. В процессе рассмотрения будет частично затронут и протокол ARP (Address Resolution Protocol), позволяющий преобразовывать IP-адреса (сетевой уровень) в 6 байтные MAC-адреса сетевых карт Ethernet (канальный уровень):

1. Сеть состоит из отдельных сегментов (подсетей), которые соединены между собой либо маршрутизаторами, либо обычными компьютерами, на которых функции маршрутизации выполняются операционной системой. Такие компьютеры имеют несколько сетевых карт, каждая из которых имеет свой адрес в соответствующей подсети и являются шлюзами (gateway) из одной подсети в другую. Шлюзом называется любое сетевое оборудование с несколькими сетевыми интерфейсами и осуществляющее продвижение пакетов между сетями на уровне протоколов сетевого уровня.
2. Адресация в сетях идет по протоколу IP, поэтому компьютер-отправитель знает IP-адрес получателя. Но для доставки пакета на аппаратном уровне необходимо знать Ethernet-адрес сетевой карты получателя. Для этого по протоколу ARP посылается широковещательное сообщение всем компьютерам в данном сегменте сети. Все компьютеры получают его, но только компьютер с указанным IP-адресом "отзывается" и сообщает Ethernet-адрес своей сетевой карты. Компьютер отправитель кэширует ответ в своей памяти и в дальнейшем (пока кэш не будет очищен) будет направлять пакеты по этому Etehrnet-адресу. Таким образом, доставка в рамках одного сегмента сети происходит напрямую.
3. Однако компьютер-получатель может и не находится в одном сегменте с отправителем (что видно по маске подсети). В таком случае, сообщение будет послано на маршрутизатор (IP-адрес маршрутизатора (шлюза) устанавливается вручную при настройке сети), который, получив широковещательный ARP-запрос, сообщит компьютеру-адресату свой Ethernet-адрес и дальнейшая связь будет идти через маршрутизатор. Маршрутизатор анализирует свои таблицы маршрутизации, и на основании их принимает решение о маршруте доставки пакета. Таблицы маршрутизации частично составляются вручную администратором сети, а частично динамически обновляются, на основании данных соседних маршрутизаторов, по протоколам RIP, OSPF, NLSP, BGP и др. Таблица маршрутизации упрощенно выглядит следующим образом (в различных операционных системах и моделях маршрутизаторов возможны различные варианты):

Таблица

Пример таблицы маршрутизации.

	Адрес назначения (сеть или компьютер)	Маска подсети	Адрес следующего маршрутизатора (шлюза)	Метрика (расстояние до адресата)	Сетевой интерфейс
1	127.0.0.1	255.255.255.255	*	1	lo
2	210.1.1.0	255.255.255.0	*	1	eth0
3	130.30.0.0	255.255.0.0	*	1	eth1
4	190.55.0.0	255.255.0.0	*	1	eth2
5	170.10.0.0	255.255.0.0	130.30.10.5	1	eth1
6	13.1.10.17	255.255.255.255	130.30.10.5	1	eth1
7	200.15.1.0	255.255.255.0	130.30.10.7	1	eth1
8	200.15.1.0	255.255.255.0	190.55.15.1	3	eth2
9	0.0.0.0	0.0.0.0	231.1.1.5	1	ppp0

Данный пример составлен для компьютера (выполняющего функции шлюза и маршрутизатора), который подключен к сети 210.1.1.0 через сетевую карту eth0, имеет связь с Internet через модем (интерфейс ppp0), а также подключен к сети 130.30.0.0 1 через сетевую карту eth1, и к сети 190.55.0.0 через сетевую карту eth2 (см. рис).

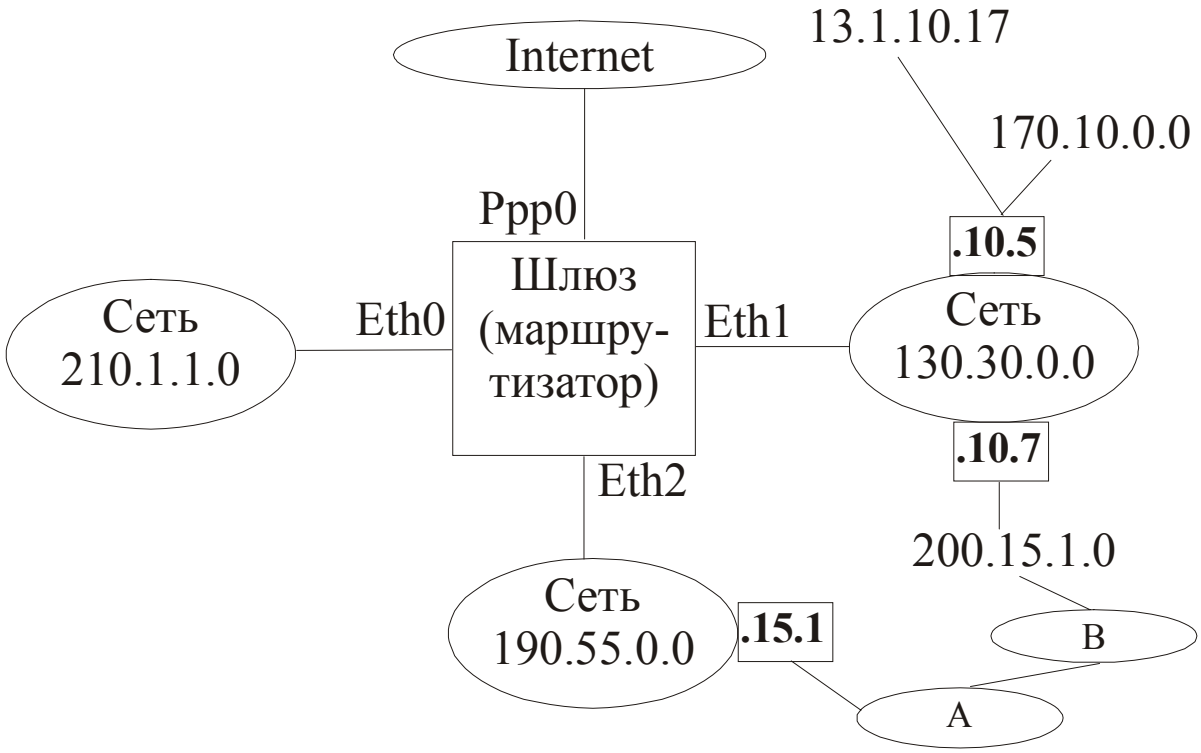


рис. Условная сеть для пояснения таблицы маршрутизации

Таблица

Пояснения к таблице маршрутизации

№ стр.	Пояснения
1	Описан loopback-адрес 127.0.0.1, т.е. ссылка на самого себя (фиктивный сетевой интерфейс lo).
2 - 4	Описываются сети, к которым непосредственно подключенные сетевые карты шлюза. Сеть 210.1.1.0 – к интерфейсу eth0, 130.30.0.0 – к интерфейсу eth1, 190.55.0.0 – к интерфейсу eth2. Доставка пакетов в эти сети происходит напрямую, поэтому в таблице адрес следующего маршрутизатора (шлюза) для них не указан.
5	Описан маршрут до сети 170.10.0.0. Все пакеты для компьютеров с адресами от 170.10.0.1 до 170.10.255.254 будут доставлены на маршрутизатор, описанный в таблице маршрутизатор с адресом 130.30.10.5 на интерфейсе Eth1.
6	Описан путь до единственного компьютера с адресом 13.1.10.17. Пакеты до него будут также доставляться на маршрутизатор 130.30.10.5 (интерфейс Eth1).

Составитель: Ляхевич А.Г., 2000 - 2002 год

Продолжение табл.

№ стр.	Пояснения
7-8	Описаны два маршрута до сети 200.15.1.0, один из которых проходит через маршрутизатор 130.30.10.7 (интерфейс eth1), а второй – через маршрутизатор 190.55.15.1 (интерфейс eth2). Маршрут, проходящий через маршрутизатор 190.55.15.1 длиннее (метрика 3) и проходит через 3 сети: сеть 190.55.0.0, сеть А и сеть В. Указанный в таблице маршрутизации маршрутизатор 190.55.15.1 является лишь промежуточным: получив пакет до сети 200.15.1.0, он, в соответствии с собственной таблицей маршрутизации, передаст пакет маршрутизатору сети А. Тот проанализирует свою таблицу маршрутизации и передаст пакет маршрутизатору сети В, который на основании своей таблицы маршрутизации доставит пакет до сети назначения 200.15.1.0. Такая цепочечная схема доставки характерна для крупных сетей и позволяет не хранить на первом маршрутизаторе 190.55.15.1 информацию о всем пути следования пакета: достаточно только знать адрес ближайшего маршрутизатора на пути к адресату. Т.е. информация распределена между большим числом маршрутизаторов в сети. В противном случае, пришлось бы на каждом маршрутизаторе хранить все пути до всех существующих сетей, что нерационально, а в сети Internet и невозможно.
9	Описан маршрут по умолчанию 0.0.0.0. Любые пакеты до сетей (компьютеров) для которых не существует записи в таблице маршрутизации будут направлены по этому маршруту, т.е. в данном случае направлены в Internet, на маршрутизатор 231.1.1.5, соединенный с данным маршрутизатором по модему (сетевой интерфейс ppp0).

Необходимо также обратить внимание на поле "метрика" таблицы маршрутизации. Обычно метрика увеличивается на 1 при прохождении каждого маршрутизатора и соответствует реальному расстоянию до сети назначения, однако для особо перегруженных маршрутов маршрутизатор может быть вручную настроен так, чтобы увеличивать метрику более чем на 1, искусственно делая маршрут более длинным. В результате пакеты, если это возможно, будут направляться по другому, более короткому маршруту, и только пакеты для которых этот маршрут является единственным (или короче всех остальных) будут направлены на этот маршрутизатор.

4. Если в процессе доставки пакета возникнет ошибка, то будет получено сообщение по протоколу ICMP, указывающего причину ошибки. По протоколу ICMP может быть передана управляющая информация, позволяющая изменить маршрут доставки на более оптимальный или вообще поменять его, если какой-то шлюз временно не работает. Протокол ICMP также позволяет посылать короткие служебные пакеты (ping), которые позволяют протестировать работоспособность сети. Если с компьютера А будет послан ping компьютеру В, то операционная система компьютера В также ответит коротким пакетом по протоколу ICMP. После получения этого пакета компьютер А во-первых знает, что компьютер В доступен, а во вторых знает за какое время пакет дошел до компьютера В и вернулся обратно. После отправки нескольких ping-ов собирается статистика: минимальное, максимальное и среднее время приема-передачи пакетов, процент утерянных пакетов. Основные виды ICMP сообщений перечислены в таблице

Таблица

Основные виды ICMP сообщений.

Тип	Код	Сообщение
0	0	Echo Reply (Эхо-ответ)
3		Destination Unreachable (Адресат недостижим по различным причинам):
	0	Net Unreachable (нет маршрута в сеть)
	1	Host Unreachable (хост недоступен)
	2	Protocol Unreachable (протокол недоступен)
	3	Port Unreachable (порт недоступен)
	4	Datagram Too Big (необходима фрагментация, но она запрещена)
	5	Source Route Failed (невозможно выполнить опцию Source Route)
	13	Communication Administratively Prohibited (обработка дейтаграммы административно запрещена)
4	0	Source Quench (Замедление источника)
5		Redirect (выбрать другой маршрутизатор для посылки датаграмм):
	0	в данную сеть
	1	на данный хост
	2	в данную сеть с данным TOS (Type Of Service – тип обслуживания)
	3	на данный хост с данным TOS (Type Of Service – тип обслуживания)
8	0	Echo (Эхо-запрос)
9	0	Router Advertisement (Объявление маршрутизатора)

Продолжение табл.

Тип	Код	Сообщение
10	0	Router Solicitation (Запрос объявления маршрутизатора)
11		Time Exceeded (Время жизни дейтаграммы истекло)
	0	при передаче
	1	при сборке
12		Parameter problem (Ошибка в параметрах)
	0	Ошибка в IP-заголовке
	1	Отсутствует необходимая опция
13	0	Timestamp (Запрос временной метки для синхронизации часов)
14	0	Timestamp Reply (Ответ на запрос временной метки)
17	0	Address Mask Request (Запрос сетевой маски)
18	0	Address Mask Reply (Ответ на запрос сетевой маски)

2.3.2. Протоколы транспортного уровня TCP и UDP.

Протоколы транспортного уровня в стеке TCP/IP представлены двумя протоколами: TCP и UDP. Протокол TCP позволяет устанавливать виртуальный канал передачи данных между компьютерами. Канал устанавливается следующим образом:

1. Компьютер A посылает компьютеру B пакет, с установленным флагом SYN (синхронизация) и случайным числом (a)

   => SYN (a).
2. Компьютер B отвечает пакетом, с установленными флагами ACK (подтверждение), с параметром (a+1), и установленным флагом SYN и своим случайным числом (b).

   <= ACK(a+1), SYN (b)
3. Компьютер A завершает "рукопожатие" пакетом, с флагами ACK(a+1), ACK (b+1).

   => ACK (a+1), ACK (b+1)

После установления канала, программа может направлять в него данные непрерывным потоком, как на стандартное устройство ввода вывода. Протокол TCP сам разобьет данные на пакеты, при помощи алгоритма "скользящего окна" обеспечит подтверждение факта получения пакетов принимающей стороной и повторную передачу пакетов, если в этом будет необходимость. Кроме того, в протоколе TCP реализованы достаточно сложные механизмы регулирования загрузки сети и устранения заторов в сети. Протокол UDP более быстр, чем протокол TCP, однако менее надежен. Данные передаются без установления виртуального канала, в предположении, что принимающая сторона ждет данные. Программа должна сама позаботиться о разбитии передаваемых данных на пакеты, протокол не содержит средств подтверждения факта доставки сообщения и средств коррекции ошибок - все эти задачи должна решать программа.

При рассмотрении протоколов транспортного уровня необходимо остановиться на понятии "порт" и "сокет". Порт в протоколах транспортного уровня – это не физически существующий порт ввода-вывода (как, например, последовательный порт COM1), а "виртуальный" порт, который программно изолирует данные передаваемые по одному порту, от данных передаваемых по другому порту. Порты нумеруются от 0 до 65535. Существуют общеизвестные порты (well known ports), каждый из которых традиционно связан с тем или иным видом сетевого приложения. Например, стандартным портом для Web-сервера является порт 80. Большинство общеизвестных портов имеют номера меньше 1024. Это связано с тем, что в ОС Unix порты с номерами меньше 1024 доступны только приложениям с привилегиями суперпользователя root (администратор), поэтому пользователь без этих привилегий не сможет запустить собственный Web-сервер, который подменит на 80 порту настоящий Web-сервер. Порты TCP и порты UDP не зависят друг от друга. Порт 80 TCP может быть занят одним сетевым приложением, а 80 порт UDP – другим приложением.

Сокет (socket) – это описатель сетевого соединения между двумя сетевыми приложениями, которое включает в себя:

- IP-адрес и номер порта локальной машины.
- IP-адрес и номер порта удаленной машины.

Сокет однозначно описывает сетевое соединение. У двух различных соединений хотя бы один из приведенных выше параметров должен отличаться. Например к 80 порту сервера могут одновременно подключиться два приложения, работающие с различных портов на клиентской машине.

3. Протоколы прикладного уровня HTTP, FTP, SMTP, IMAP, POP3, TELNET.

В соответствии с архитектурой клиент-сервер, программа делится на две части (одна работает на сервере, вторая – на компьютере пользователя), функционирующие как единое целое. Протоколы прикладного уровня описывают взаимодействие клиентской и серверной частью программы. Выделяют следующие наиболее известные прикладные протоколы:

1. HTTP (Hyper Text Transfer Protocol)- протокол передачи гипертекста, работает на 80 порту. Используется в WWW для передачи гипертекстовых HTML страниц. При работе по этому протоколу, каждый элемент HTML – страницы загружается отдельно, причем соединение между загрузками прерывается и никакой информации о соединении не сохраняется. Это сделано для того, чтобы пользователя Web-страниц каждый получал "по чуть-чуть, в порядке общей очереди". В противном случае могла бы создаться ситуация, когда один человек качает страницу с большим количеством рисунков высокого разрешения, а все остальные ждут пока он это закончит.
2. FTP (File Transfer Protocol.) – протокол передачи файлов, работает на 20 и 21 порту. Предназначен для копирование файлов между компьютерами. Полностью занимает канал, пока не будет получен файл, сохраняет информацию о соединении. При сбое возможна докачка с того места, где произошел сбой.
3. SMTP, IMAP-4, POP3 – почтовые протоколы (электронная почта). SMTP - 25 порт, IMAP-4 – 143 порт, POP3 – 110 порт. Отличие: SMTP – протокол расчитанный на доставку почты до конкретного получателя, POP3 и IMAP-4 – протоколы взаимодействия пользователя со своим почтовым ящиком на сервере. При использовании SMTP предполагается, что почтовый адрес указывает на компьютер конечного получателя, и на этом компьютере запущена специальная программа, которая принимает и обрабатывает почту. Однако чаще всего бывает, что почта не доставляется на компьютер каждого отдельного пользователя, а обрабатывается централизованно, на отдельном почтовом сервере. В таком случае, каждый пользователь имеет на почтовом сервере свой почтовый ящик. Почта доставляется до сервера по протоколу SMTP (конечный получатель – сервер) и помещается в почтовые ящики пользователей. Затем пользователи подключаются к своим почтовым ящикам по протоколу POP3 или IMAP-4 и забирают почту. Протокол POP3 требует полностью скачать себе всю почту, а затем разбираться: нужна она вам была или нет. Причем, чаще всего, администратор запрещает хранить копии скачанной почты на сервере (или ограничивает время хранения копий), поэтому, например, скачав почту из почтового ящика на институтский компьютер, вы полностью очистите свой почтовый ящик и, зайдя на почтовый ящик с домашнего компьютера, увидите сообщение "Писем нет". Протокол IMAP-4 позволяет просматривать на сервере заголовки писем (указывается статус письма: новое, отвеченное и т.п.) и скачивать с сервера только необходимые письма или даже часть некоторого письма. Также можно на стороне сервера проводить поиск по сообщениям, создавать иерархию каталоговдля хранения полученных писем (копии скачанных писем остаются на сервере, пока вы их не удалите). Фактически IMAP4 дублирует функции почтовых программ пользователя (например, Microsoft Outlook), однако существенной разницей здесь являеется то, что если Micrsoft Outlook работает на компьютере пользователя, то команды протокола IMAP-4 выполняются на сервере, а значит каталоги с письмами хранятся в одном месте (на сервере), что очень удобно если вы часто подключаетесь к серверу с разных компьютеров и не хотите на каждом компьютере иметь полную копию всех писем.

   Резюмируя вышесказанное можно привести наиболее распространенный вариант работы с почтой для обычного пользователя: отправка почты – по протоколу SMTP (на почтовый сервер получателя), получение почты – по протоколу POP3 или IMAP-4 (скачивание почты из почтового ящика на своем почтовом сервере).
4. TELNET – используется для подключения и управления удаленным компьютером, работает на 23 порту. После подключения каждый символ, введенный на локальной машине, обрабатывается так, как если бы он был введен на удаленной машине. Либо может использоваться командный режим – управление удаленной машиной при помощи специальных команд. Фактически TELNET – это протокол эмуляции терминала: при помощи TELNET можно подключиться, например, на 25 порт и вручную набрать все необходимые поля заголовка письма, изменив адрес отправителя (обычно эти поля заполняются автоматически специальными почтовыми программами) и отправить само письмо. Или, например, подключиться на 80 порт и "поиграть" роль Web-браузера Internet Explorer.

4. Система доменных имен DNS.

Доменное имя – это имя компьютера, вида www.sait.com. Адресация в Internet происходит по IP-адресам, однако для человека гораздо удобнее доменные имена.

> *Существует также термин URL-адрес (Universal Resource Locator), т.е. запись вида http://www.sait.com, или в полном варианте http://www.sait.com:80/katalog/index.html#glava1. Доменное имя являеется частью URL-адреса (схема_передачи:// доменное_имя : порт / имя файла# внутреняя_ссылка).*

Встает проблема: как поставить в соответствие IP-адрес и доменное имя. Вести на каждом компьютере базу данных, содержащую все доменные имена Internet, невозможно, поэтому применяется служба доменных имен DNS (Domain Name Service). Алгоритм ее функционирования таков:

1. Пользователь в окне Web-браузера вводит http://www.microsoft.com
2. На первый DNS сервер IP-адрес которого известен (устанавливается в настройке Windows вручную или автоматически провайдером, при подключении к нему) компьютером пользователя направляется запрос на установление IP-адреса по доменному имени.

3. Если в базе данных сервера имеется соответствующая запись доменное имя – IP адрес, то ответ в виде IP-адреса возвращается компьютеру пользователю. Если в базе данных информация отсутствует, то запрос передается на DNS – сервер более высокого уровня (его IP известен серверу), который скорее всего тоже не знает ответ, но зато знает какой DNS сервер более низкого уровня отвечает за данную зону доменных имен, и перенаправит запрос ему. Тот ответит, и запрос по цепочке вернется к компьютеру пользователя. Такая схема наиболее распространена, однако возможна и другая схема. Если в базе данных сервера отсутствует запрашиваемая запись доменное имя – IP адрес, то компьютеру пользователя будет возвращен IP-адрес DNS-сервера более высокого уровня, и компьютер пользователя должен впоследствии сам выполнять запросы к последующим DNS-серверам.

Нет однозначного соответствия между IP–адресом и доменным именем. Компьютер, имеющий один и тот же IP-адрес, может иметь доменное имя www.minsk.by www.usa.com nowhere.ru и т.д. Для этого достаточно купить доменное имя, т.е. заплатить за регистрацию соответствующего IP-адреса в базе данных DNS – серверов, отвечающих за соответсвующие зоны имен. При этом сам компьютер может физически находиться хоть в Китае, или вообще, весь сайт может реально находиться на сервере, предоставляющем бесплатное размещение web-страниц (web-хостинг), в каталоге www.halyava.fi /pub/web/sait/5873, но вы купили доменное имя www.kruto.by и теперь пользователи могут попасть на ваш сайт, используя это имя.

За каждую зону имен отвечает минимум два DNS-сервера. Записи базы данных DNS-сервера хранятся в файле зоны, в формате, определяемом стандартом RFC-1035. Существует несколько типов записей для хранения раз личных данных. Рассмотрим эти записи подробнее.

Запись типа SOA.

SOA (Start of Authority) - начало зоны. Первая запись в базе данных зоны. Пример:

```
exmpl.ru. IN SOA  ns.exmpl.ru.  hostmaster.ns.exmpl.ru. (
        1997120802
        10800
        3600
        3600000
        86400 )
```

Таблица

Расшифровка полей записи SOA

Поле	Значение
exmpl.ru.	Полностью уточненное имя зоны. Полностью уточненное доменное имя должно оканчиваться точкой. Если в файле зоны какое-либо другое имя не заканчивается на точку, то к данному имени добавляется имя зоны, т.е. имя server читается, как server.exmpl.ru.
SOA	Начало зоны
ns.exmpl.ru.	Доменное имя первичного DNS-сервера зоны.
hostmaster.ns.exmpl.ru.	Адрес электронной почты администратора DNS-сервера. Символ @ заменяется на точку, т.е. данный адрес выглядит, как hostmaster@ns.exmpl.ru.
1997120802	Серийный номер версии данных (выбор номера произволен, но номер должен увеличиваться после каждой модификации данных; обычно используется формат *YYYYMMDDVV,* где *YYYY* —год, *MM*— месяц, *DD* — день и *VV* — порядковый номер модификации зоны в указанный день).
10800	Период запросов на обновление данных со стороны вторичного сервера (в секундах).
3600	период повторов попыток запроса данных вторичным сервером в случае неудачи первого запроса (в секундах)
3600000	срок годности данных, то есть время, через которое вторичный сервер прекратит обслуживать запросы, если ему не удастся восстановить связь с первичным сервером (в секундах)
86400	время жизни данных зоны в кэше запросившего их сервера (в секундах)

Запись типа NS.

NS (Name Server) — указание серверов DNS, обслуживающих запросы к данной зоне имен. Указывается как первичный, так и вторичные сервера DNS. Пример:

```
exmpl.ru.        IN  NS      ns.exmpl.ru.
                 IN  NS      server.eldorado.com.
```

Таблица

Расшифровка полей записи NS

Поле	Значение
exmpl.ru.	Имя зоны, для которой указываются обслуживающие ее DNS-сервера.
NS	Сервер имен, т.е. DNS-сервер.
ns.exmpl.ru.	Доменное имя DNS-сервера зоны. Данный DNS-сервер являеется первичным сервером, что следует из записи SOA.
server.eldorado.com.	Доменное имя DNS-сервера зоны. Данный DNS-сервер являеется вторичным сервером. Для повышения надежности рекомендуется, чтобы сервера DNS находились в разных IP-сетях (не путать IP-сеть с зоной имен DNS).

Запись типа A.

A (Address) — указание IP-адреса для соответствующего доменного имени. Для одного и того же IP адреса может быть определено несколько доменных имен (например, на одном компьютере работает несколько Web-серверов, или один и тот же Web-сервер имеет несколько имен). Одно и то же доменное имя может иметь несколько записей типа A (это соответствует случаю, когда один узел имеет несколько IP-интерфейсов). Пример:

```
ns      IN  A   1.16.195.2
mail    IN  A   1.16.195.1
wildcat IN  A   1.16.195.3
```

Так как имена в примере не полностью уточненные, то к ним добавляется имя зоны **exmpl.ru,** то есть эти записи эквивалентны следующим:

```
ns.exmp1.ru.        IN  A   1.16.195.2
mail.exmpl.ru.      IN  A   1.16.195.1
wildcat.exmpl.ru.   IN  A   1.16.195.3
```

Запись типа CNAME.

CNAME (Canonical Name) — определение псевдонимов для доменных имен. В примере, приведенном ниже, доменное имя www.exmpl.ru. соответствует имени wildcat.exmpl.ru. Пример:

```
www     IN      CNAME       wildcat.exmpl.ru.
```

Запись типа MX.

MX (Mail Exchanger) — указание почтового сервера. Очень часто для обработки почты выделяется отдельный почтовый сервер, который получает, обрабатывает и хранит почту пользователей. Пример:

```
pochta.exmpl.ru.        IN  MX      10      mai1.exmp1.ru.
                        IN  MX      20      wildcat.exmpl.ru.
specpochta.exmpl.ru.    IN  MX      10      wildcat.exmpl.ru.
```

Таблица

Расшифровка полей записи MX

Поле	Значение
pochta.exmpl.ru.	Имя почтового домена, для которого указывается обслуживающий его почтовый сервер. Все почтовые сообщения, адресованные на адреса вида ящик@pochta.exmpl.ru. , будут отправляться для обработки на почтовые сервера, указанные ниже. Самого компьютера, с именем pochta.exmpl.ru., может и не существовать вообще.
MX	Почтовый сервер.
10	Приоритет почтового сервера. Одно и то же доменное имя может иметь несколько записей MX, указывающих на разные обработчики почты. Сначала будет предпринята попытка доставить почту на сервер с наименьшим числом в поле приоритета (в нашем случае, на mai1.exmp1.ru.). Если связь с данным сервером установить не удастся, то почта будет доставлена на компьютер wildcat.exmp1.ru., где она будет находится до тех пор, пока связь с компьютером mai1.exmp1.ru. не будет восстановлена. После восстановления связи, почта будет доставлена на mail.exmpl.ru. Компьютер wildcat.exmpl.ru. в данном случае выступает просто как почтовый ретранслятор, что необходимо для повышения надежности доставки почты.
mail.exmpl.ru.	Почтовый сервер для почтового домена pochta.exmpl.ru.

Продолжение табл.

Поле	Значение
wildcat.exmpl.ru.	Почтовый ретранслятор (mail relay) для почтового домена pochta.exmpl.ru.
specpochta.exmpl.ru.	Имя почтового домена. Все почтовые сообщения, адресованные на адреса вида ящик@specpochta.exmpl.ru. будут направлены на обработку серверу wildcat.exmpl.ru.

При отсутствии записи MX для какого-либо доменного имени почта, адресованная на это доменное имя, будет доставляться непосредственно на хост, имеющий такое имя. Только каноническое доменное имя (не псевдоним) может быть указано в качестве обработчика почты.

Пример файла базы данных зоны exmpl.ru.

```
exmpl.ru.       IN      SOA     ns.exmpl.ru.    hostmaster.ns.exmpl.ru. (
                                1997120802
                                10800
                                3600
                                3600000
                                86400 )
exmpl.ru.                       IN      NS              ns.exmpl.ru.
                                IN      NS              server.eldorado.com.
pochta.exmpl.ru.                IN      MX   10         mail.exmpl.ru.
                                IN      MX   20         wildcat.exmpl.ru.
specpochta.exmpl.ru.            IN      MX   10         wildcat.exmpl.ru.
mail                            IN      A               1.16.195.1
wildcat                         IN      A               1.16.195.3
ns                              IN      A               1.16.195.2
www                             IN      CNAME           wildcat.exmpl.ru.
```

Файл базы данных обратной зоны

Если файл базы данных прямой зоны служит для преобразования доменных имен в IP-адреса, то файл базы данных обратной зоны используется для преобразования IP-адресов в доменные имена. Название обратной зоны строится по следующему принципу: в обратном порядке записывается сетевая часть IP-адреса, к которой добавляется стандартный домен in-addr.arpa. Например, для сети 1.16.195.0 обратная зона будет иметь название 195.16.l.in-addr.arpa. В файле базы данных обратной зоны используются следующие типы записей.

Запись типа PTR

PTR (Pointer) — указатель на доменное имя. В примере, приведенном ниже, указывается, что IP-адрес 1.16.195.1 соответствует доменному имени ns.exmpl.ru. Пример (для зоны 195.16.l.in-addr.arpa):

```
98      IN      PTR     ns.exmpl.ru.
```

Или, полностью уточняя *имя* в левой части:

Запись типа A

Запись типа A имеет специальное значение для имен из домена in-addr.arpa и указывает маску подсети. Пример (в базе данных зоны 195.16.1 .in-addr.arpa):

```
0       IN      A       255.255.255.0
```

Пример базы данных обратной зоны

```
195.16.1.in-addr.arpa.  IN      SOA     ns.exmpl.ru.    hostmaster.ns.exmpl.ru. (
                                1997120802
                                10800
                                3600
                                3600000
                                86400 )
        IN      NS      ns.exmpl.ru.
        IN      NS      server.eldorado.com.
0       IN      A       255.255.255.0
1       IN      PTR     mail.exmpl.ru.
2       IN      PTR     ns.exmpl.ru.
3       IN      PTR     wildcat.exmpl.ru.
```