**Министерство образования и науки Российской Федерации**
**Нижнекамский химико-технологический институт (филиал)**
Федерального государственного бюджетного образовательного учреждения
высшего профессионального образования
«Казанский национальный исследовательский технологический университет»

**С.А. Мерзляков, Д.В. Елизаров**

# СЕТИ ЭВМ В СИСТЕМАХ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И ПРОИЗВОДСТВ

## ТЕКСТЫ ЛЕКЦИЙ

**Нижнекамск**
**2013**

**УДК 681.5**
**М 52**

Печатаются по решению редакционно-издательского совета Нижнекамского химико-технологического института (филиала) ФГБОУ ВПО «КНИТУ».

**Рецензенты:**
**Долганов А.В.,** кандидат технических наук;
**Гусев С.Н.,** руководитель группы по созданию ИУС, ОАО «ТАНЕКО»

**Мерзляков, С.А.**
**М 52** Сети ЭВМ в системах автоматизации технологических процессов и производств : тексты лекций / С.А. Мерзляков, Д.В. Елизаров. – Нижнекамск : Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2013. – 115с.

Тексты лекций соответствуют требованиям федерального государственного образовательного стандарта направлений бакалаврской подготовки 220400 «Управление в технических системах», 230100 «Информатика и вычислительная техника».

Приведены основные теоретические аспекты проектирования и создания компьютерных локальных сетей в системах автоматизации технологических процессов и производств.

Тексты лекций предназначены для студентов всех форм обучения факультета управления и автоматизации.

Подготовлены на кафедре автоматизации технологических процессов и производств Нижнекамского химико-технологического института (филиал) ФГБОУ ВПО «Казанский национальный исследовательский технологический университет».

**УДК 681.5**

© Мерзляков С.А., Елизаров Д.В., 2013
© Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2013

## Содержание

**1.1.**

# 1. Общие принципы построения сетей

## 1.1. Простейшая сеть из двух компьютеров

Главной целью объединения компьютеров в сеть было разделение ресурсов: пользователи компьютеров, подключенных к сети, или приложения, выполняемые на этих компьютерах, получают возможность доступа к ресурсам компьютеров сети, к таким как:

- периферийные устройства, такие как диски, принтеры, плоттеры, сканеры и др.;
- данные, хранящиеся в оперативной памяти или на внешних запоминающих устройствах;
- вычислительная мощность.

## 1.2. Сетевые интерфейсы

Для связи устройств в них, прежде всего, должны быть предусмотрены внешние интерфейсы.

Интерфейс — в широком смысле — формально определенная логическая и/или физическая граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

Разделяют физический и логический интерфейсы.

- Физический интерфейс (называемый также портом) — определяется набором электрических связей и характеристиками сигналов. Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение.
- Логический интерфейс (называемый также протоколом) — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.

Интерфейс компьютер—компьютер позволяет двум компьютерам обмениваться информацией. С каждой стороны он реализуется парой:

- аппаратным модулем, называемым сетевым адаптером, или сетевой интерфейсной картой;
- драйвером сетевой интерфейсной карты — специальной программой, управляющей работой сетевой интерфейсной карты.

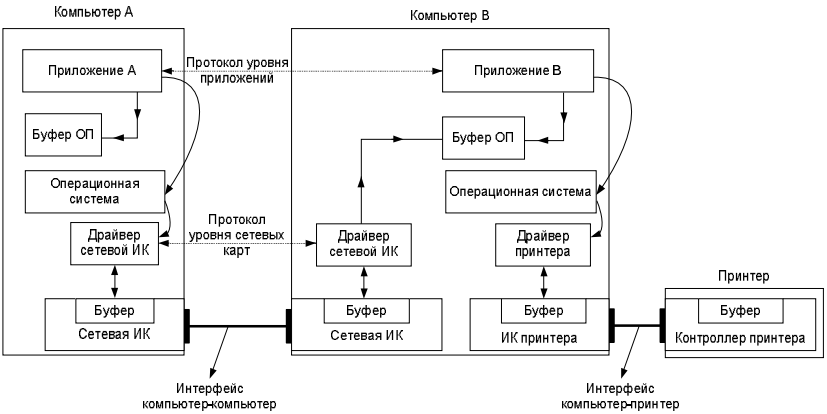



**Рис. 1** Совместное использование принтера в компьютерной сети

Интерфейс компьютер—периферийное устройство (в данном случае интерфейс компьютер—принтер) позволяет компьютеру управлять работой периферийного устройства (ПУ). Этот интерфейс реализуется:

- со стороны компьютера — интерфейсной картой и драйвером ПУ (принтера), подобным сетевой интерфейсной карте и ее драйверу;
- со стороны ПУ — контроллером ПУ (принтера), обычно представляющий собой аппаратное устройство, принимающее от компьютера как данные, например байты информации, которую нужно распечатать на бумаге, так и команды, которые он отрабатывает, управляя электромеханическими частями периферийного устройства, например, выталкивая лист бумаги из принтера или перемещая магнитную головку диска.

# 2. Проблемы связи нескольких компьютеров

## 2.1. Топология физических связей

Объединяя в сеть несколько (больше двух) компьютеров, необходимо решить, каким образом соединить их друг с другом, иначе, выбрать конфигурацию физических связей, или топологию.

Под топологией сети понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а

рёбрам — физические или информационные связи между вершинами.

Компьютеры можно соединять друг с другом последовательно, предполагая, что они будут передавать сообщения «транзитом». В качестве транзитного узла может выступать как универсальный компьютер, так и специализированное устройство.

От выбора топологии связей существенно зависят характеристики сети:

- наличие между узлами нескольких путей повышает надежность сети и делает возможным распределение нагрузки между отдельными каналами;
- простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой;
- экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи.

Среди множества возможных конфигураций различают полносвязные и неполносвязные.

Полносвязная топология соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Этот вариант оказывается громоздким и неэффективным. В таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов. Полносвязные топологии в крупных сетях применяются редко. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

Все другие варианты основаны на неполносвязных топологиях, когда для обмена данными между двумя компьютерами может потребоваться транзитная передача данных через другие узлы сети.

**1. Кольцевая топология.** Данные передаются по кольцу от одного компьютера к другому. Главным достоинством кольца является то, что оно по своей природе обеспечивает резервирование связей. Данные в кольце, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату. Это свойство используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца.

**2. Звездообразная топология.** Она образуется в случае, когда каждый компьютер подключается непосредственно к общему

центральному устройству, называемому концентратором. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство.

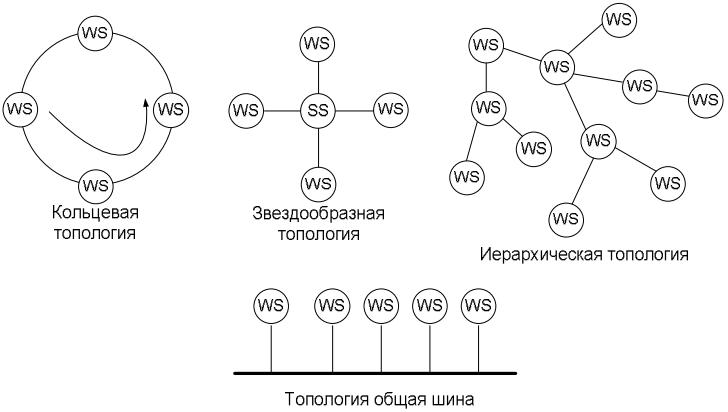



**Рис. 2** Типовые топологии сетей

Недостатки звездообразной топологии:

- более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства;
- возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

**3. Иерархическая топология.** Иногда необходимо построить сеть с использованием нескольких концентраторов, иерархически соединенных между собой звездообразными связями. Получаемую в результате структуру называют иерархической звездой, или деревом. В настоящее время дерево является самой распространенной топологией связи, как в локальных, так и глобальных сетях.

**4. Топология общая шина.** Особым частным случаем звезды является общая шина. Здесь в качестве центрального элемента выступает пассивный кабель (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Достоинства: дешевизна и простота присоединения новых узлов к сети, а недостатки: низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая

производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

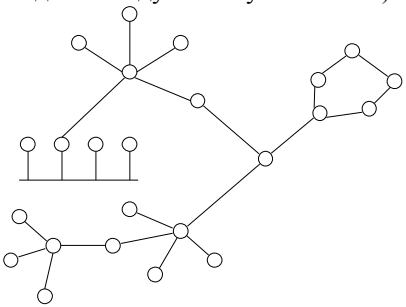


**Рис. 3** Смешанная топология

**5. Смешанная топология.** Небольшие сети имеют типовую топологию - звезда, кольцо или общая шина. Поскольку для крупных сетей характерно наличие произвольных связей между компьютерами, то в них можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со смешанной топологией.

## 2.2. Адресация узлов сети

Одной из проблем, которую нужно учитывать при объединении трех и более компьютеров, является проблема адресации, а именно адресации их сетевых интерфейсов. Один компьютер может иметь несколько сетевых интерфейсов. Например, для создания полносвязной структуры из *N* компьютеров необходимо, чтобы у каждого из них имелся *N-1* интерфейс.

По количеству адресуемых интерфейсов адреса можно классифицировать следующим образом:

- уникальный адрес (unicast) используется для идентификации отдельных интерфейсов;
- групповой адрес (multicast) идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;
- широковещательный адрес (broadcast), пакеты с таким адресом должны быть доставлены всем узлам сети;
- адрес произвольной рассылки (anycast), определенный в новой версии протокола IPv6, так же, как и групповой адрес, задает

группу адресов, однако данные, посланные по этому адресу, должны быть доставлены не всем адресам данной группы, а любому из них.

Адреса могут быть числовыми (например, 129.26.255.255 или 81.la.ff.ff) и символьными (site.domen.ru).

Символьные адреса (имена) удобны для восприятия человеком и поэтому обычно несут смысловую нагрузку.

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется адресным пространством.

Адресное пространство может иметь плоскую (линейную) организацию или иерархическую организацию.

При плоской организации множество адресов никак не структурировано. Примером плоского числового адреса является MAC-адрес, который предназначен для однозначной идентификации сетевых интерфейсов в локальных сетях. Такой адрес обычно используется аппаратурой и записывается в виде двоичного или шестнадцатеричного числа, например 00:1A:6C:16:2B:1C. MAC-адреса встраиваются в аппаратуру компанией-изготовителем, поэтому их называют также аппаратными адресами (hardware address).

При иерархической организации адресное пространство структурируется в виде вложенных друг в друга подгрупп, которые, последовательно сужая адресуемую область, в конце концов, определяют отдельный сетевой интерфейс.

Представителями такого типа адресов являются сетевые IP- и IPX-адреса. В них поддерживается двухуровневая иерархия: адрес делится на старшую часть — номер сети и младшую — номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла требуется уже после доставки сообщения в нужную сеть. На практике обычно применяют сразу несколько схем адресации, так что сетевой интерфейс компьютера может одновременно иметь несколько адресов-имен. Каждый адрес задействуется в той ситуации, когда соответствующий вид адресации наиболее удобен. А для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, которые называют протоколами разрешения адресов.

## 2.3. Коммутация

Пусть компьютеры физически связаны между собой в соответствии с некоторой топологией.

Соединение конечных узлов через сеть транзитных узлов называют коммутацией. Последовательность узлов, лежащих на пути от отправителя к получателю, образует маршрут.

В качестве примера рассмотрим рис. 4.

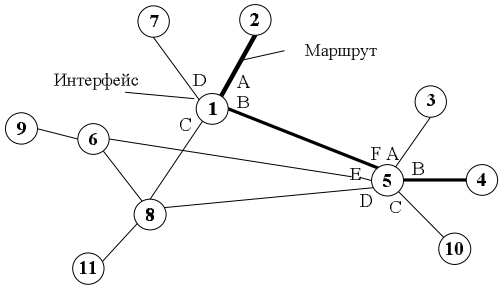



**Рис. 4** Коммутация абонентов через сеть транзитных узлов

Здесь несвязанные непосредственно между собой узлы *2* и *4*, вынуждены передавать данные через транзитные узлы, в качестве которых могут выступить, например, узлы *1* и *5*. Узел *1* должен выполнить передачу данных между своими интерфейсами *A* и *B*, а узел *5* — между интерфейсами *F* и *B*. В данном случае маршрутом является последовательность: *2-1-5-4*, где *2* — узел-отправитель, *1* и *5* — транзитные узлы, *4* — узел-получатель.

## 2.4. Обобщенная задача коммутации

В общем виде задача коммутации может быть представлена в виде следующих взаимосвязанных частных задач:

**1.** Определение информационных потоков, для которых требуется прокладывать маршруты.

**2.** Маршрутизация потоков.

**3.** Продвижение потоков, то есть распознавание потоков и их локальная коммутация на каждом транзитном узле.

**4.** Мультиплексирование и демультиплексирование потоков.

## 2.5. Маршрутизация

Задача маршрутизации, в свою очередь, включает в себя две подзадачи:

- определение маршрута;
- оповещение сети о выбранном маршруте.

Определить маршрут означает выбрать последовательность транзитных узлов и их интерфейсов, через которые надо передавать данные, чтобы доставить их адресату.

Определение маршрута — сложная задача, особенно когда конфигурация сети такова, что между парой взаимодействующих сетевых интерфейсов существует множество путей. Чаще всего выбор останавливают на одном оптимальном по некоторому критерию маршруте. В качестве критериев могут выступать, например, номинальная пропускная способность и загруженность каналов связи; задержки, вносимые каналами; количество промежуточных транзитных узлов; надежность каналов и транзитных узлов.

Маршрут может определяться эмпирически («вручную») администратором сети, но этот подход неудобен для большой сети со сложной топологией. В этом случае используются автоматические методы определения маршрутов. Для этого конечные узлы и другие устройства сети оснащаются специальными программными средствами, которые организуют взаимный обмен служебными сообщениями, позволяющий каждому узлу составить свое «представление» о сети. Затем на основе собранных данных программными методами определяются рациональные маршруты.

При выборе маршрута часто ограничиваются только информацией о топологии сети. Этот подход иллюстрирует рис. 5. Для передачи трафика между конечными узлами *A* и *C* существует два альтернативных маршрута: *A-1-2-3-C* и *A-1-3-C*. Если мы учитываем только топологию, то выбор очевиден — маршрут *A-1-3-C,* который имеет меньше транзитных узлов.

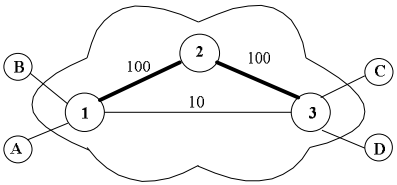



**Рис. 5** Выбор маршрута

## 2.6. Продвижение данных

Итак, пусть маршруты определены, записи о них сделаны в таблицах всех транзитных узлов, все готово к передаче данных между абонентами (коммутации абонентов).

Тогда, прежде всего, отправитель должен выставить данные на тот свой интерфейс, с которого начинается найденный маршрут, а все

транзитные узлы должны соответствующим образом выполнить «переброску» данных с одного своего интерфейса на другой, другими словами, выполнить коммутацию интерфейсов.

Устройство, функциональным назначением которого является коммутация, называется коммутатором. На рис. 6 показан коммутатор, который переключает информационные потоки между четырьмя своими интерфейсами.

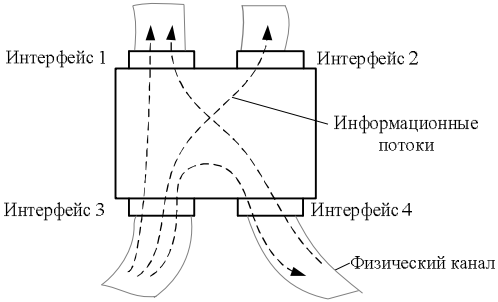



**Рис. 6** Коммутатор

Коммутатором может быть как специализированное устройство, так и универсальный компьютер со встроенным программным механизмом коммутации. В этом случае коммутатор называется программным.

## 2.7. Мультиплексирование и демультиплексирование

Чтобы определить, на какой интерфейс следует передать поступившие данные, коммутатор должен выяснить, к какому потоку они относятся. Эта задача должна решаться независимо от того, поступает на вход коммутатора только один «чистый» поток или «смешанный» поток.

Демультиплексирование — разделение суммарного агрегированного потока на несколько составляющих его потоков.

Мультиплексирование — образование из нескольких отдельных потоков общего агрегированного потока, который передается по одному физическому каналу связи.

Другими словами, мультиплексирование — это способ разделения одного имеющегося физического канала между несколькими одновременно протекающими сеансами связи между абонентами сети.

Одним из основных способов мультиплексирования потоков является разделение времени. При этом способе каждый поток время от времени (с фиксированным или случайным периодом) получает в своё

полное распоряжение физический канал и передает по нему свои данные.

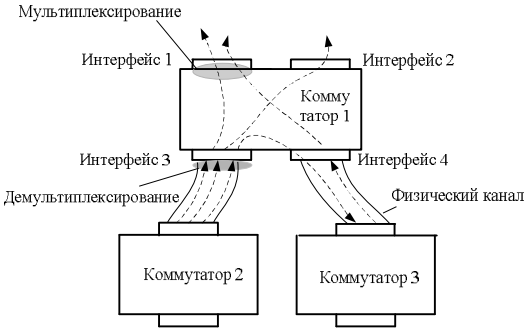



**Рис. 7** Операции мультиплексирования и демультиплексирования потоков при коммутации

Также распространено частотное разделение канала - когда каждый поток передаёт данные в выделенном ему частотном диапазоне.

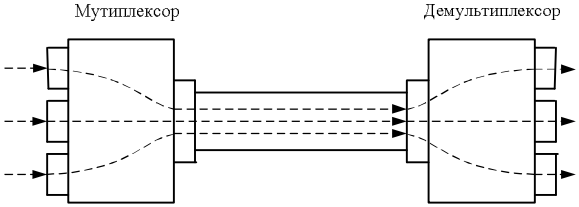



**Рис. 8** Мультиплексор и демультиплексор

## 2.8. Типы коммутации

Среди множества возможных подходов к решению задачи коммутации выделяют два основополагающих, к которым относят коммутацию каналов и коммутацию пакетов.

## 3. Коммутация каналов

Коммутация каналов — организация составного канала через несколько транзитных узлов из нескольких последовательно «соединённых» каналов на время передачи сообщения (оперативная коммутация) или на более длительный срок.

## 3.1. Элементарный канал

Элементарный канал (или просто канал) — это базовая техническая характеристика сети с коммутацией каналов, представляющая собой некоторое фиксированное в пределах данного типа сетей значение пропускной способности. Любая линия связи в сети с коммутацией каналов имеет пропускную способность, кратную элементарному каналу, принятому для данного типа сети.

Значение элементарного канала, или, другими словами, минимальная единица пропускной способности линии связи, выбирается с учетом разных факторов. Но элементарный канал не стоит выбирать меньше минимально необходимой пропускной способности.

Особенностью сетей с коммутацией каналов является то, что пропускная способность каждой линии связи должна быть равна целому числу элементарных каналов.

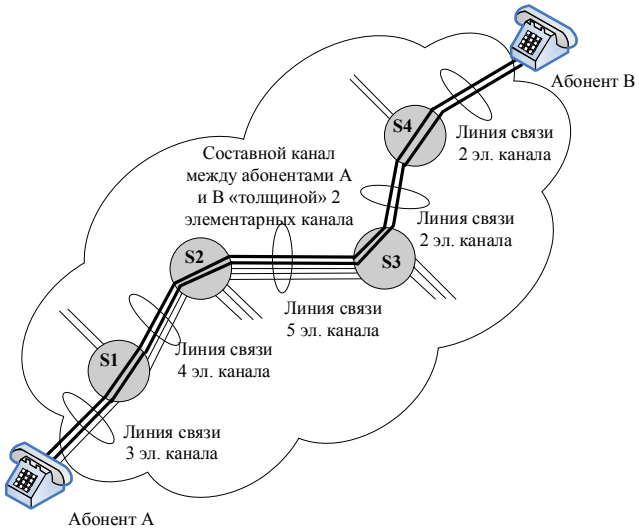



**Рис. 9** Составной канал в сети с коммутацией каналов

На рис. 9 представлен фрагмент сети. Предположим, что эта сеть характеризуется элементарным каналом *P* бит/с. В сети существуют линии связи разной пропускной способности, состоящие из 2, 3, 4 и 5 элементарных каналов. На рисунке показаны два абонента, *A* и *B*, генерирующие во время сеанса связи (телефонного разговора)

информационный поток, для которого в сети был предусмотрен маршрут, проходящий через четыре коммутатора *S1*, *S2*, *S3* и *S4*. Предположим также, что интенсивность информационного потока между абонентами не превосходит *2P* бит/с. Тогда для обмена данными этим двум абонентам достаточно иметь в своем распоряжении по паре элементарных каналов, «выделенных» из каждой линии связи, лежащей на маршруте следования данных от пункта *A* к пункту *B*. На рисунке эти элементарные каналы, необходимые абонентам *A* и *B*, обозначены толстыми линиями.

## 3.2. Составной канал

Связь, построенную путем коммутации (соединения) элементарных каналов, называют составным каналом.

Свойства составного канала:

- составной канал на всем своем протяжении состоит из одинакового количества элементарных каналов;
- составной канал имеет постоянную и фиксированную пропускную способность на всем своем протяжении;
- составной канал создается временно на период сеанса связи двух абонентов;
- на время сеанса связи все элементарные каналы, входящие в составной канал, поступают в исключительное пользование абонентов, для которых был создан этот составной канал;
- в течение всего сеанса связи абоненты могут посылать в сеть данные со скоростью, не превышающей пропускную способность составного канала;
- данные, поступившие в составной канал, гарантированно доставляются вызываемому абоненту без задержек, потерь и с той же скоростью (скоростью источника) вне зависимости от того, существуют ли в это время в сети другие соединения или нет;
- после окончания сеанса связи элементарные каналы, входившие в соответствующий составной канал, объявляются свободными и возвращаются в пул распределяемых ресурсов для использования другими абонентами.

В сети может одновременно происходить несколько сеансов связи (обычная ситуация для телефонной сети, в которой одновременно передаются разговоры сотен и тысяч абонентов). Разделение сети между сеансами связи происходит на уровне элементарных каналов. Например (см. рис. 9), мы можем предположить, что после того как в

линии связи *S2 - S3* было выделено два канала для связи абонентов *A* и *B*, оставшиеся три элементарных канала были распределены между тремя другими сеансами связи, проходившими в это же время и через эту же линию связи. Такое мультиплексирование позволяет одновременно передавать через каждый физический канал трафик нескольких логических соединений.

Таким образом, продвижение данных в сетях с коммутацией каналов происходит в два этапа:

**1.** в сеть поступает служебное сообщение — запрос, который несёт адрес вызываемого абонента и организует создание составного канала;

**2.** по подготовленному составному каналу передаётся основной поток данных, для передачи которого уже не требуется никакой вспомогательной информации, в том числе адреса вызываемого абонента.

Коммутация данных в коммутаторах выполняется на основе локальных признаков — номеров элементарных каналов.

Запросы на установление соединения не всегда завершаются успешно. Если на пути между вызывающим и вызываемым абонентами отсутствуют свободные элементарные каналы или вызываемый узел занят, то происходит отказ в установлении соединения.

## 3.3. Неэффективность при передаче пульсирующего трафика

Сети с коммутацией каналов наиболее эффективно передают пользовательский граф в том случае, когда скорость его постоянна в течение всего сеанса связи и максимально соответствует фиксированной пропускной способности физических линий связи сети. Эффективность работы сети снижается, когда информационные потоки, генерируемые абонентами, приобретают пульсирующий характер.

Так, разговаривая по телефону, люди постоянно меняют темп речи, сменяя быстрые высказывания паузами. В результате соответствующие «голосовые» информационные потоки становятся неравномерными, а значит, снижается эффективность передачи данных. В случае телефонных разговоров это снижение оказывается вполне приемлемым и позволяет широко использовать сети с коммутацией каналов для передачи голосового трафика.

Гораздо сильнее снижает эффективность сети с коммутацией

каналов передача трафика, генерируемого приложениями, с которыми работает пользователь компьютера. Этот трафик практически всегда является пульсирующим. Если для описанного сеанса доступа в Интернет вы задействуете сеть с коммутацией каналов, то большую часть времени составной канал между вашим компьютером и веб-сервером будет простаивать. В то же время часть производительности сети окажется закреплённой за вами и останется недоступной другим пользователям сети.

Для эффективной передачи неравномерного компьютерного трафика была специально разработана техника коммутации пакетов.

## 4. Коммутация пакетов

Сети с коммутацией пакетов, так же как и сети с коммутацией каналов, состоят из коммутаторов, связанных физическими линиями связи. Однако передача данных в этих сетях происходит совершенно по-другому.

Сеть с коммутацией пакетов не создает заранее для своих абонентов отдельных, выделенных исключительно для них каналов связи. Данные могут задерживаться и даже теряться по пути следования.

Важнейшим принципом функционирования сетей с коммутацией пакетов является представление информации, передаваемой по сети, в виде структурно отделенных друг от друга порций данных, называемых пакетами.

Каждый пакет снабжен заголовком, в котором содержится адрес назначения и другая вспомогательная информация (длина поля данных, контрольная сумма и др.), используемая для доставки пакета адресату. Наличие адреса в каждом пакете является одним из важнейших особенностей техники коммутации пакетов, так как каждый пакет может быть обработан коммутатором независимо от других пакетов, составляющих сетевой трафик. Помимо заголовка у пакета имеется концевик. В концевике обычно помещается контрольная сумма, которая позволяет проверить, была ли искажена информация при передаче через сеть или нет.

Пакеты поступают в сеть без предварительного резервирования линий связи и не с фиксированной заранее заданной скоростью. Предполагается, что сеть с коммутацией пакетов, в отличие от сети с коммутацией каналов, всегда готова принять пакет от конечного узла.

Как и в сетях с коммутацией каналов, в сетях с коммутацией пакетов для каждого из потоков вручную или автоматически

определяется маршрут, фиксируемый в хранящихся на коммутаторах таблицах коммутации. Пакеты, попадая на коммутатор, обрабатываются и направляются по тому или иному маршруту на основании информации, содержащейся в их заголовках, а также в таблице коммутации (рис. 10).

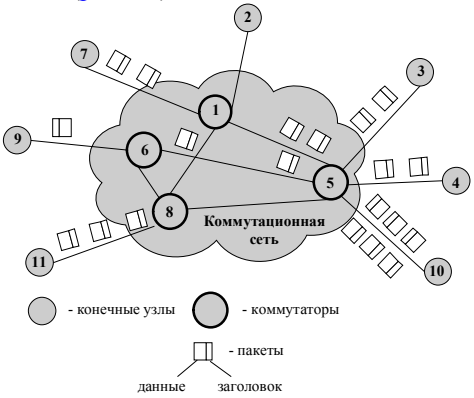



**Рис. 10** Передача данных по сети в виде пакетов

Пакеты, принадлежащие как одному и тому же, так и разным информационным потокам, при перемещении по сети могут «перемешиваться» между собой, образовывать очереди и «тормозить» друг друга. На пути пакетов могут встретиться линии связи, имеющие разную пропускную способность. В зависимости от времени суток может сильно меняться и степень загруженности линий связи. В таких условиях не исключены ситуации, когда пакеты, принадлежащие одному и тому же потоку, могут перемещаться по сети с разными скоростями и даже прийти к месту назначения не в том порядке, в котором они были отправлены.

Разделение данных на пакеты позволяет передавать неравномерный компьютерный трафик более эффективно, чем в сетях с коммутацией каналов. Это объясняется тем, что пульсации трафика от отдельных компьютеров носят случайный характер и распределяются во времени так, что их пики чаще всего не совпадают. Поэтому когда линия связи передает трафик большого количества конечных узлов, то в суммарном потоке пульсации сглаживаются, и пропускная способность линии используется более рационально, без длительных простоев. Этот эффект иллюстрируется рис. 11, на котором показаны неравномерные потоки пакетов, поступающие от конечных узлов 3,4 и 10 в сети, изображенной на рис. 10.

Важным аспектом всей системы передачи пакетов, является буферизация пакетов (рассмотреть самостоятельно).

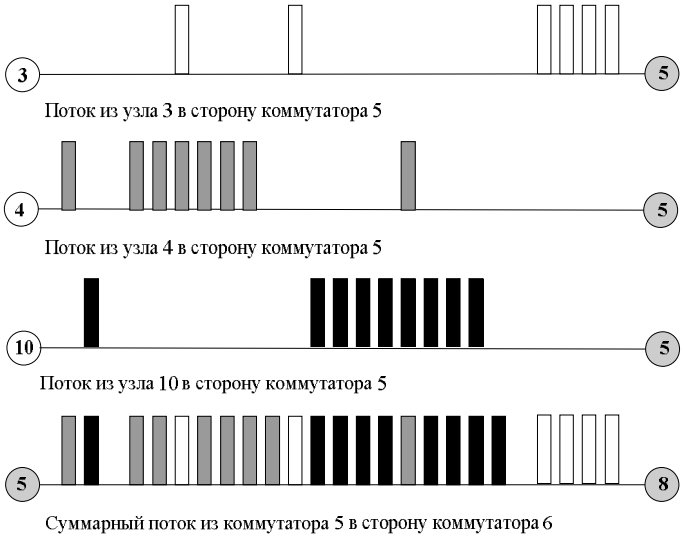



**Рис. 11** Сглаживание трафика в сетях с коммутацией пакетов

# 5. Способы соединения абонентов в сети

## 5.1. Дейтаграммная передача

Дейтаграммный способ передачи данных основан на том, что все передаваемые пакеты продвигаются (передаются от одного узла сети другому) независимо друг от друга на основании одних и тех же правил.

Решение о продвижении пакета принимается на основе таблицы коммутации, ставящей в соответствие адресам назначения пакетов информацию, однозначно определяющую следующий по маршруту транзитный (или конечный) узел. В качестве такой информации могут выступать идентификаторы интерфейсов данного коммутатора или адреса входных интерфейсов коммутаторов, следующих по маршруту.

На рис. 12 показана сеть, в которой шесть конечных узлов (*N1-N6*) связаны семью коммутаторами (*S1-S7*), несколько перемещающихся по разным маршрутам пакетов с разными адресами назначения (*N1-N6*), на пути которых лежит коммутатор *S1*.

При поступлении каждого из этих пакетов в коммутатор *S1*

выполняется просмотр соответствующей таблицы коммутации и выбор дальнейшего пути перемещения. Так пакет с адресом *N5* будет передан коммутатором *S1* на интерфейс, ведущий к коммутатору *S6*, где в результате подобной процедуры этот пакет будет направлен конечному узлу-получателю *N5*.

| Адрес назначения | Адрес следующего коммутатора |
|---|---|
| N1 | Пакет не требуется передавать через сеть |
| N2 | S2 |
| N3 | S3 |
| N4 | S3 |
| N5 | S6 |
| N6 | S6 |

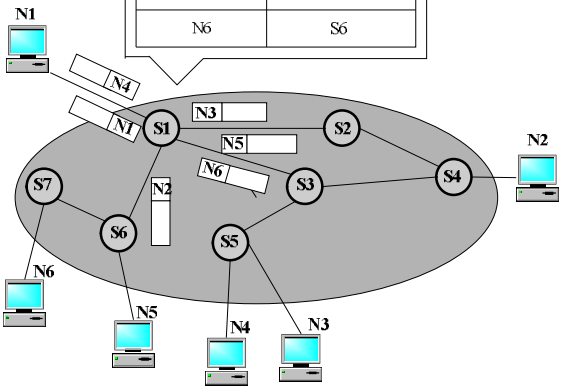



**Рис. 12** Иллюстрация дейтаграммного принципа передачи пакетов

## 5.2. Передача с установлением логического соединения

Процедура согласования двумя конечными узлами сети некоторых параметров процесса обмена пакетами называется установлением логического соединения. Параметры, о которых договариваются два взаимодействующих узла, называются параметрами логического соединения.

Наличие логического соединения позволяет более рационально по сравнению с дейтаграммным способом обрабатывать пакеты. Например, при потере нескольких предыдущих пакетов может быть снижена скорость отправки последующих. Благодаря нумерации пакетов и отслеживанию номеров отправленных и принятых пакетов можно повысить надежность путем отбрасывания дубликатов.

Осуществить упорядочивание поступивших и повторение передачи потерянных пакетов.

Параметры соединения могут быть: постоянными, то есть не изменяющимися в течение всего соединения (например, идентификатор соединения, способ шифрования пакета или максимальный размер поля данных пакета), или переменными, то есть динамически отражающими текущее состояние соединения (например, последовательные номера передаваемых пакетов).

Логическое соединение может быть рассчитано на передачу данных как в одном направлении - от инициатора соединения, так и в обоих направлениях. После передачи некоторого законченного набора данных, например определенного файла, узел-отправитель инициирует разрыв данного логического соединения, посылая соответствующий служебный кадр.

После того как соединение установлено и все параметры согласованы, конечные узлы начинают передачу собственно самих данных. Пакеты данных обрабатываются коммутаторами точно так же, как и при дейтаграммной передаче: из заголовков пакетов извлекаются адреса назначения и сравниваются с записями в таблицах коммутации, содержащих информацию о следующих шагах по маршруту. Так же как дейтаграммы, пакеты, относящиеся к одному логическому соединению, в некоторых случаях (например, при отказе линии связи) могут доставляться адресату по разным маршрутам.

Передача с установлением соединения предоставляет больше возможностей в плане надежности и безопасности обмена данными, чем дейтаграммная передача. Однако, этот способ более медленный, так как он подразумевает дополнительные вычислительные затраты на установление и поддержание логического соединения.

## 5.3. Передача с установлением виртуального канала

Единственный заранее проложенный фиксированный маршрут, соединяющий конечные узлы в сети с коммутацией пакетов, называют виртуальным каналом (virtual circuit или virtual channel).

Виртуальные каналы прокладываются для устойчивых информационных потоков. Для того чтобы выделить поток данных из общего трафика, каждый пакет этого потока помечается специальной меткой.

В одной и той же сетевой технологии могут быть задействованы разные способы продвижения данных. Так, дейтаграммный протокол

IP используется для передачи данных между отдельными сетями, составляющими Интернет. В то же время обеспечением надежной доставки данных между конечными узлами этой сети занимается протокол TCP, устанавливающий логические соединения без фиксации маршрута. И наконец, Интернет — это пример сети, применяющей технику виртуальных каналов, так как в состав Интернета входит немало сетей ATM и Frame Relay, поддерживающих виртуальные каналы.

## 6. Сравнение сетей с коммутацией пакетов и каналов

Два коммутатора объединены каналом связи с пропускной способностью 100 Мбит/с. Пользователи подключаются к сети с помощью каналов доступа (access link) с пропускной способностью 10 Мбит/с. Предположим, что все пользователи создают одинаковый пульсирующий трафик со средней скоростью 1 Мбит/с. При этом в течение непродолжительных периодов времени скорость предложенной нагрузки возрастает до максимальной скорости канала доступа, то есть до 10 Мбит/с. Такие периоды длятся не более одной секунды. Предположим также, что все пользователи, подключенные к коммутатору *S1*, передают информацию только пользователям, подключенным к коммутатору *S2*.

Представленная на рисунке сеть является сетью с коммутацией каналов. Поскольку скорость пользовательского трафика достигают 10 Мбит/с, каждому из пользователей необходимо установить соединение с пропускной способностью 10 Мбит/с. Таким образом, одновременно через сеть смогут передавать данные только 10 пользователей. Суммарная средняя скорость передачи данных через сеть будет равна только 10 Мбит/с (10 пользователей передают данные со средней скоростью 1 Мбит/с). Следовательно, линия связи между коммутаторами, хотя и имеет общую пропускную способность 100 Мбит/с, используется только на 10 %.

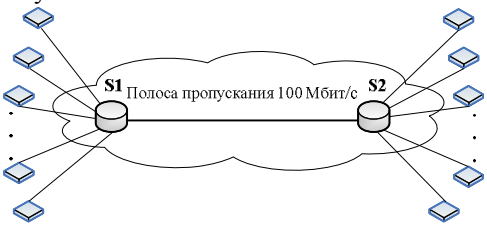



**Рис. 13** Сравнение эффективности сетей с коммутацией пакетов и каналов

Рассмотрим вариант, когда та же сеть работает на основе техники коммутации пакетов. При средней скорости пользовательских потоков 1 Мбит/с сеть может передавать одновременно до 100/1 = 100 информационных потоков пользователей, полностью расходуя пропускную способность канала между коммутаторами. Однако, это справедливо, если емкости буферов коммутаторов достаточно для хранения пакетов во время перегрузки, когда суммарная скорость потока данных превышает 100 Мбит/с. Оценим необходимый объем буфера коммутатора *S1*. За период перегрузки в коммутатор *S1* от каждого потока поступит *10 Мбит/с * 1 с =10 Мбит*, а от 100 потоков — 1000 Мбит. Из этих данных за одну секунду коммутатор успеет передать в выходной канал только 100 Мбит. Значит, чтобы ни один пакет не был потерян во время перегрузки сети, общий объем входных буферов коммутатора должен быть не меньше *1000-100=900 Мбит*, или более 100 Мбайт. Современные коммутаторы обычно имеют меньшие объемы буферов (1-10 Мбайт).

На основании данных таблицы 1 можно определить, в каких случаях рациональнее использовать сети с коммутацией каналов, а в каких — с коммутацией пакетов.

**Таблица 1**. Сравнение сетей с коммутацией каналов и пакетов

| Коммутация каналов | Коммутация пакетов |
|---|---|
| Необходимо предварительно устанавливать соединение | Отсутствует этап установления соединения (дейтаграммный способ) |
| Адрес требуется только на этане установления соединения | Адрес и другая служебная информация передаются с каждым пакетом |
| Сеть может отказать абоненту в установлении соединения | Сеть всегда готова принять данные от абонента |
| Гарантированная пропускная способность (полоса пропускания) для взаимодействующих абонентов | Пропускная способность сети для абонентов неизвестна, задержки передачи носят случайный характер |
| Трафик реального времени передается без задержек | Ресурсы сети используются эффективно при передаче пульсирующего трафика |
| Высокая надежность передачи | Возможные потери данных из-за переполнения буферов |
| Нерациональное использование пропускной способности каналов, снижающее общую эффективность сети | Автоматическое динамическое распределение пропускной способности физического канала между абонентами |

# 7. Архитектура и стандартизация сетей

## 7.1. Протокол и стек протоколов

В процессе обмена сообщениями участвуют, по меньшей мере, две стороны, то есть необходимо организовать согласованную работу двух иерархий аппаратных и программных средств на разных компьютерах. Оба участника сетевого обмена должны принять множество соглашений.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется стеком протоколов.

## 7.2. Общая характеристика модели OSI

В модели OSI средства взаимодействия делятся на семь уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический (рис. 14). Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

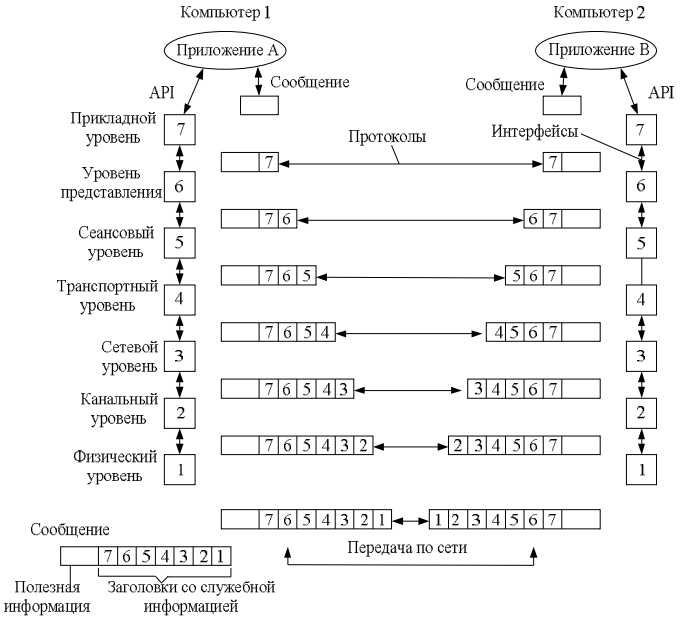



**Рис. 14** Модель взаимодействия открытых систем ISO/OSI

Приложения могут реализовывать собственные протоколы взаимодействия, используя для этих целей многоуровневую совокупность системных средств. Именно для этого в распоряжение программистов предоставляется прикладной программный интерфейс (Application Program Interface, API). В соответствии с идеальной схемой модели OSI приложение может обращаться с запросами только к самому верхнему уровню — прикладному, однако на практике многие стеки коммуникационных протоколов предоставляют возможность программистам напрямую обращаться к сервисам, или службам, расположенным на нижних уровнях.

Пусть приложение узла *A* взаимодействует с приложением узла *B*. Для этого приложение *A* обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата.

После формирования сообщения прикладной уровень направляет его вниз по стеку уровня представления. Протокол уровня представления на основании информации, полученной из заголовка сообщения прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который, в свою очередь, добавляет свой заголовок и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце в виде так называемого концевика.) Наконец, сообщение достигает нижнего, физического, уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 15).

Физический уровень помещает сообщение на физический выходной интерфейс компьютера 1, и оно начинает свое «путешествие» по сети (до этого момента сообщение передавалось от одного уровню другому в пределах компьютера 1).

Когда сообщение по сети поступает на входной интерфейс компьютера 2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Протоколы каждого из уровней между собой непосредственно не

общаются, в этом общении всегда участвуют посредники — средства протоколов нижележащих уровней. И только физические уровни различных узлов взаимодействуют непосредственно.

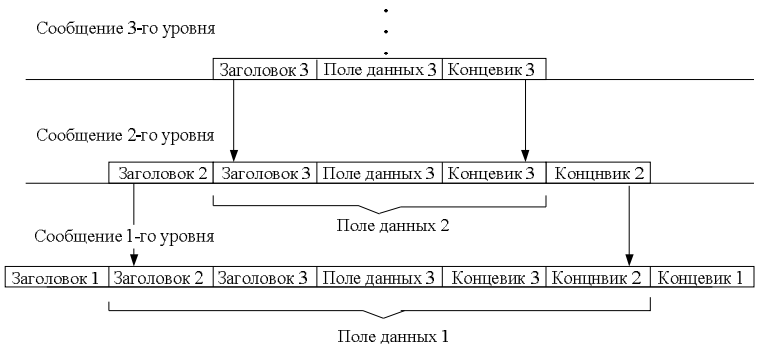



**Рис. 15** Вложенность сообщений различных уровней

## 7.3. Физический уровень

Физический уровень (physical layer) имеет дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или радиосреда. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Физический уровень не вникает в смысл информации, которую он передает. Для него эта информация представляет собой однородный поток битов, которые нужно доставить без искажений и в соответствии с заданной тактовой частотой (интервалом между соседними битами).

## 7.4. Канальный уровень

Канальный уровень (data link layer) обеспечивает прозрачность

соединения для сетевого уровня. Для этого он предлагает ему следующие услуги:

1. Установление логического соединения между взаимодействующими узлами.
2. Согласование скоростей передатчика и приемника информации.
3. Обеспечение надежной передачи, обнаружение и коррекция ошибок.

Для решения поставленных задач канальный уровень из пакетов формирует собственные протокольные единицы данных — кадры, состоящие из поля данных и заголовка. Канальный уровень помещает пакет в поле данных одного или нескольких кадров и заполняет собственной служебной информацией заголовок кадра.

В сетях, построенных на основе разделяемой среды, канальный уровень проверяет доступность разделяемой среды. Эту функцию иногда выделяют в отдельный подуровень управления доступом к среде (Medium Access Control, MAC).

Протоколы канального уровня реализуются как на конечных узлах (средствами сетевых адаптеров и их драйверов), так и на всех промежуточных сетевых устройствах.

## 7.5. Сетевой уровень

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой составной сетью, или Интернетом.

Технология, позволяющая соединять в единую сеть множество сетей, в общем случае построенных на основе разных технологий, называется технологией межсетевого взаимодействия (Internetworking).

На рис. 16 показано несколько сетей, каждая из которых использует собственную технологию канального уровня: Ethernet, FDDI, Token Ring, ATM, Frame Relay.

Чтобы связать между собой сети, построенные на основе столь отличающихся технологий, нужны дополнительные средства, такие средства предоставляет сетевой уровень.

Функции сетевого уровня реализуются:

- группой протоколов;
- специальными устройствами — маршрутизаторами.

Одной из функций маршрутизатора является физическое соединение сетей. Маршрутизатор имеет несколько сетевых

интерфейсов, подобных интерфейсам компьютера, к каждому из которых может быть подключена одна сеть. Таким образом, все интерфейсы маршрутизатора можно считать узлами разных сетей. Маршрутизатор может быть реализован программно на базе универсального компьютера (например, типовая конфигурация Unix или Windows включает программный модуль маршрутизатора). Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня.

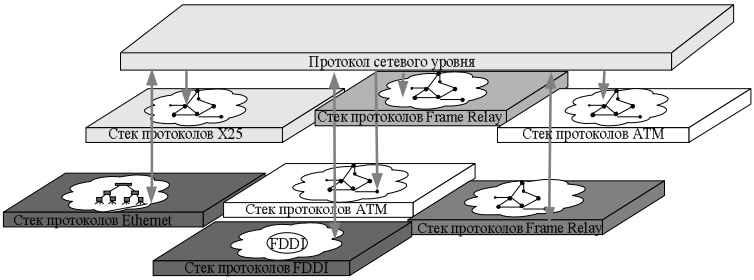



**Рис. 16** Необходимость сетевого уровня

Определение маршрута является важной задачей сетевого уровня. Маршрут описывается последовательностью сетей (или маршрутизаторов), через которые должен пройти пакет, чтобы попасть к адресату.

Сетевой уровень для решения своей задачи обращается к нижележащему канальному уровню. Весь путь через составную сеть разбивается на участки от одного маршрутизатора до другого, причем каждый участок соответствует пути через отдельную сеть.

Для того чтобы передать пакет через очередную сеть, сетевой уровень помещает его в поле данных кадра соответствующей канальной технологии, указывая в заголовке кадра канальный адрес интерфейса следующего маршрутизатора. Сеть, используя свою канальную технологию, доставляет кадр по заданному адресу. Маршрутизатор извлекает пакет из прибывшего кадра и после необходимой обработки передает пакет для дальнейшей транспортировки в следующую сеть, предварительно упаковав его в новый кадр канального уровня в общем случае другой технологии. Таким образом, сетевой уровень играет роль координатора, организующего совместную работу сетей, построенных на основе разных технологий.

## 7.6. Транспортный уровень

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов транспортного сервиса от низшего класса 0 до высшего класса 4. Эти виды сервиса отличаются качеством предоставляемых услуг, срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью обнаружения и исправления ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/ IP.

## 7.7. Сеансовый уровень

Сеансовый уровень (session layer) управляет взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса. Эти средства позволяют в ходе длинных передач сохранять информацию о состоянии этих передач в виде контрольных точек, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов. Функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

## 7.8. Уровень представления

Уровень представления (presentation layer) обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть

синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб.

## 7.9. Прикладной уровень

Прикладной уровень (application layer) — это набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением.

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. К протоколам прикладного уровня относится, в частности, упоминавшийся ранее протокол HTTP, с помощью которого браузер взаимодействует с веб-сервером. Приведем в качестве примера также несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP, SMB в Microsoft Windows, NCP в операционной системе Novell NetWare.

# 8. Понятие открытой системы

Открытой может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

Под термином «спецификация» в вычислительной технике понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, особых характеристик. Понятно, что не всякая спецификация является стандартом.

Под открытыми спецификациями понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.

Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а

также создавать программно-аппаратные комплексы из продуктов разных производителей.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы. Например, открытость семейства операционных систем Unix заключается, помимо всего прочего, в наличии стандартизованного программного интерфейса между ядром и приложениями, что позволяет легко переносить приложения из среды одной версии Unix в среду другой версии.

Модель OSI касается только одного аспекта открытости, а именно — открытости средств взаимодействия устройств, связанных в компьютерную сеть. Здесь под открытой системой понимается сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами по стандартным правилам, определяющим формат, содержание и значение принимаемых и отправляемых сообщений.

Если две сети построены с соблюдением принципов открытости, это дает следующие преимущества:

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;
- безболезненная замена отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами;
- легкость сопряжения одной сети с другой.

Ярким примером открытой системы является Интернет.

## 9. Распределение протоколов по элементам сети

На рис. 17 показаны основные элементы компьютерной сети: конечные узлы — компьютеры и промежуточные узлы — коммутаторы и маршрутизаторы.

Полный стек протоколов реализован только на конечных узлах, а промежуточные узлы поддерживают протоколы всех трех нижних уровней. Для продвижения пакетов по сети достаточно функционирование трех нижних уровней.

Сетевые повторители, работающие на физическом уровне, называются концентраторами, или хабами. Они повторяют электрические сигналы, поступившие на один из их интерфейсов, на других своих интерфейсах, улучшая их характеристики — мощность и форму сигналов, синхронность их следования.

Коммутаторы локальных сетей поддерживают протоколы двух нижних уровней, физического и канального, что дает им возможность работать в пределах стандартных топологий.

Маршрутизаторы должны поддерживать протоколы всех трех уровней, так как сетевой уровень нужен им для объединения сетей различных технологий, а протоколы нижних уровней для взаимодействия с конкретными сетями, образующими составную сеть.

В компьютерах коммуникационные протоколы всех уровней (кроме физического и части функций канального уровня) реализуются программно операционной системой или системными приложениями.

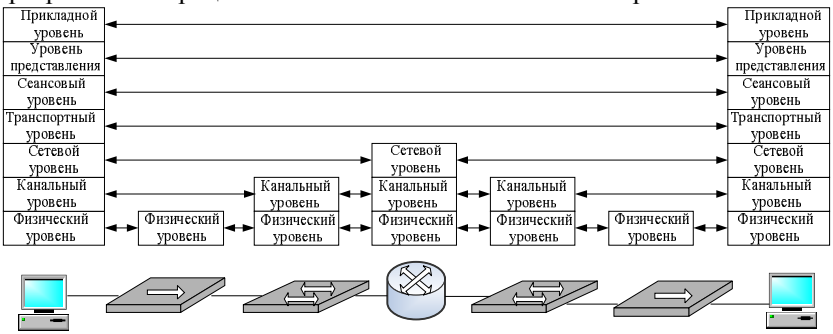



**Рис. 17** Соответствие функций различных устройств сети уровням модели OSI

Конечные узлы сети (компьютеры и компьютеризованные устройства, например мобильные телефоны) всегда предоставляют как информационные, так и транспортные услуги, а промежуточные узлы сети — только транспортные.

## 9.1. Вспомогательные протоколы транспортной системы

В реальных сетях некоторые из коммуникационных устройств поддерживают не только протоколы трех нижних уровней, но и протоколы верхних уровней. Так, маршрутизаторы реализуют протоколы маршрутизации, позволяющие автоматически строить таблицы маршрутизации, а концентраторы и коммутаторы часто поддерживают протоколы SNMP и Telnet, которые не нужны для выполнения основных функций этих устройств, но позволяют конфигурировать их и управлять ими удаленно. Все эти протоколы являются протоколами прикладного уровня и выполняют некоторые вспомогательные (служебные) функции транспортной системы.

Очевидно, что для работы прикладных протоколов сетевые устройства должны также поддерживать протоколы промежуточных уровней, таких как IP и TCP/UDP.

Вспомогательные протоколы можно разделить на группы в соответствии с их функциями:

- Первую группу вспомогательных протоколов представляют протоколы маршрутизации, такие как RIP, OSPF, BGP. Без этих протоколов маршрутизаторы не смогут продвигать пакеты, так как таблица маршрутизации будет пустой (если только администратор не заполнит ее вручную).
- Другая группа вспомогательных протоколов выполняет преобразование адресов. Здесь работает протокол DNS, который преобразует символьные имена узлов в IP-адреса. Протокол DHCP позволяет назначать IP-адреса узлам динамически, а не статически, что облегчает работу администратора сети.
- Третью группу образуют протоколы, которые используются для управления сетью. В стеке TCP/IP здесь находится протокол SNMP (Simple Network Management Protocol — простой протокол управления сетью), который позволяет автоматически собирать информацию об ошибках и отказах устройств, а также протокол Telnet, с помощью которого администратор может удаленно конфигурировать коммутатор или маршрутизатор.

## 10. Структурированная кабельная система зданий

Структурированная кабельная система здания (Structured Cabling System, SCS) — это набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методика их совместного использования, которая позволяет создавать постоянные, легко расширяемые структуры связей в вычислительных сетях. Здание состоит из этажей, а каждый этаж, в свою очередь, состоит из определенного количества комнат, соединенных коридорами. Структура здания предопределяет структуру его кабельной системы.

Структурированная кабельная система здания представляет собой своего рода «конструктор», с помощью которого проектировщик сети строит нужную конфигурацию из стандартных кабелей, соединенных стандартными разъемами и коммутируемых стандартных кроссовых панелей. При необходимости конфигурацию связей можно легко

изменить — добавить компьютер, сегмент, коммутатор, изъять ненужное оборудование, поменять соединение между компьютером и концентратором.

Наиболее детально на сегодня разработаны стандарты кабельных систем зданий, при этом иерархический подход к процессу создания такой кабельной системы позволяет назвать ее структурированной. На основе SCS работает одна или несколько локальных сетей организаций или подразделений одной организации, размещенной в этом здании. SCS планируется и строится иерархически с магистралью и многочисленными ответвлениями от неё (рис. 18).

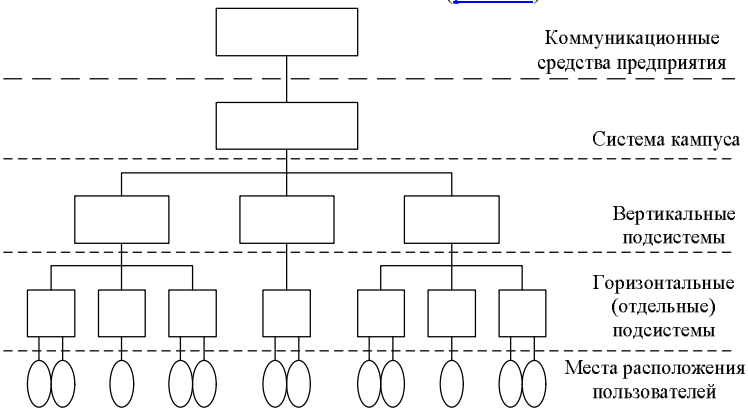



**Рис. 18** Иерархия структурной кабельной системы

Типичная иерархия SCS включает (рис. 19):

- горизонтальные подсистемы, соответствующие этажам здания — они соединяют кроссовые шкафы этажа с розетками пользователей;
- вертикальные подсистемы, соединяющие кроссовые шкафы каждого этажа с центральной аппаратной здания;
- подсистема кампуса, объединяющая несколько зданий с главной аппаратной всего кампуса (эта часть кабельной системы обычно называется магистралью).

Использование структурированной кабельной системы вместо хаотически проложенных кабелей дает предприятию ряд преимуществ:

- универсальная среда для передачи компьютерных данных в локальные вычислительные сети;
- организация локальной телефонной сети;
- передача видеоинформации;

- передача сигналов от датчиков пожарной безопасности или охранной системы.

Подобная универсализация позволяет автоматизировать многие процессы контроля, мониторинга и управления хозяйственными службами и системами жизнеобеспечения предприятия.

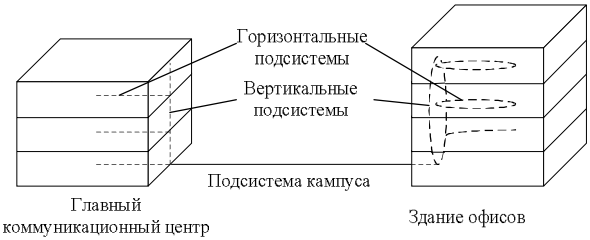



**Рис. 19** Структура кабельных подсистем

Применение SCS делает более экономичным добавление новых пользователей и изменения их мест размещения. Известно, что стоимость кабельной системы определяется в основном не стоимостью кабеля, а стоимостью работ по его прокладке. Поэтому выгоднее провести однократную работу по прокладке кабеля, возможно, с большим запасом по длине, чем несколько раз выполнять прокладку, наращивая длину кабеля.

## 11. Стандартизация протоколов локальных сетей

В институте IEEE был организован комитет 802 стандартизации технологии LAN. Результатом работы комитета IEEE 802 стало принятие семейства стандартов IEEE 802.x, содержащих рекомендации по проектированию нижних уровней локальных сетей.

Комитет IEEE 802 сегодня является основным международным органом, разрабатывающим стандарты технологий локальных сетей, в том числе коммутируемых локальных сетей, а также стандарты беспроводных локальных сетей на разделяемой среде.

Структуру стандартов IEEE 802 иллюстрирует рис. 20

Помимо индивидуальных для каждой технологии уровней существует и общий уровень, который был стандартизован рабочей группой 802.2.

Появление этого уровня связано с тем, что комитет 802 разделил функции канального уровня модели OSI на два уровня:

- управление логическим каналом (Logical Link Control, LLC);
- управление доступом к среде (Media Access Control, MAC).

Основными функциями уровня MAC являются:

- обеспечение доступа к разделяемой среде;
- передача кадров между конечными узлами посредством функций и устройств физического уровня.

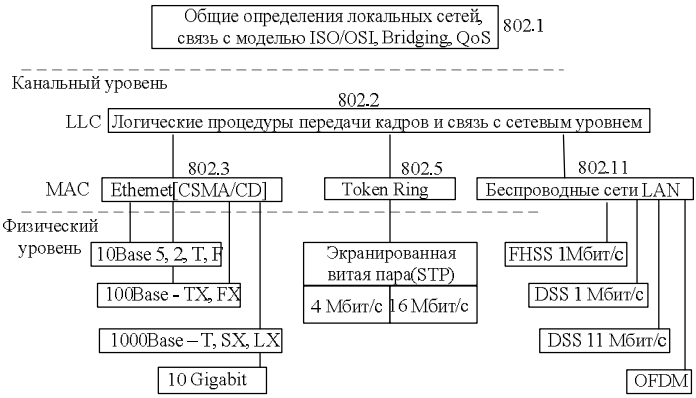



**Рис. 20** Структура стандартов IEEE 802.x

Если уровень MAC специфичен для каждой технологии и отражает различия в методах доступа к разделяемой среде, то уровень LLC представляет собой обобщение функций разных технологий по обеспечению передачи кадра с различными требованиями к надежности.

Логика образования общего для всех технологий уровня LLC заключается в следующем, после того как узел сети получил доступ к среде в соответствии с алгоритмом, специфическим для конкретной технологии, дальнейшие действия узла или узлов по обеспечению надежной передачи кадров от технологии не зависят.

Так как в зависимости от требований приложения может понадобиться разная степень надежности, то рабочая группа 802.2 определила три типа услуг:

1. Услуга LLC1 — это услуга без установления соединения и без подтверждения получения данных. LLC1 дает пользователю средства для передачи данных с минимумом издержек. В этом случае LLC поддерживает дейтаграммный режим работы, как и MAC, так что и технология LAN в целом работает в дейтаграммном режиме.

2. Услуга LLC2 дает пользователю возможность установить логическое соединение перед началом передачи любого блока данных и, если это требуется, выполнить процедуру восстановления после ошибок и упорядочивание потока блоков в рамках установленного

соединения.

3. Услуга LLC3 — это услуга без установления соединения, но с подтверждением получения данных. В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), с одной стороны, временные издержки установления логического соединения перед отправкой данных неприемлемы, а с другой стороны, подтверждение о корректности приема переданных данных необходимо.

# 12. Ethernet со скоростью 10 Мбит/с на разделяемой среде

## 12.1. MAC-адреса

На уровне MAC, который обеспечивает доступ к среде и передачу кадра, для идентификации сетевых интерфейсов узлов сети, используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые MAC-адресами. Обычно MAC-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточием, например 00-1A-2B-6C-FF-5E. Каждый сетевой адаптер имеет, по крайней мере, один MAC-адрес.

MAC-адрес также может определять группу интерфейсов или даже все интерфейсы сети. Первый (младший) бит старшего байта адреса назначения является признаком того, что адрес является индивидуальным или групповым. Если он равен 0, то он является индивидуальным, то есть идентифицирует один сетевой интерфейс, а если 1 - групповым. И в частном случае, если групповой адрес состоит из всех единиц, то есть имеет шестнадцатеричное представление OxFFFFFFFFFFFF, он идентифицирует все узлы сети и называется широковещательным.

Второй бит старшего байта адреса определяет способ назначения адреса: централизованный или локальный. Если этот бит, равен 0 (что бывает почти всегда в стандартной аппаратуре Ethernet), это говорит о том, что адрес назначен централизованно по правилам IEEE 802.

## 12.2. Форматы кадров технологии Ethernet

| 6 байт | 6 байт | 2 байта | 46–1500 байт | 4 байта |
|---|---|---|---|---|
| DA | SA | T | Данные | FCS |

**Рис. 21** Формат кадра Ethernet DIX (II)

Первые два поля заголовка отведены под адреса:

- DA (Destination Address) — MAC-адрес узла назначения;
- SA (Source Address) — MAC-адрес узла отправителя.

Для доставки кадра достаточно одного адреса — адреса назначения. Адрес источника помещается в кадр для того, чтобы узел, получивший кадр, знал, от кого пришел кадр и кому нужно на него ответить.

- Поле T (Type, или EtherType) содержит условный код протокола верхнего уровня, данные которого находятся в поле данных кадра. Это поле требуется для поддержки интерфейсных функций мультиплексирования и демультиплексирования кадров при взаимодействии с протоколами верхних уровней.
- Поле данных может содержать от 46 до 1500 байт. Если длина пользовательских данных меньше 46 байт, то это поле дополняется до минимального размера байтами заполнения.
- Поле контрольной последовательности кадра (Frame Check Sequence, FCS) состоит из 4 байт контрольной суммы. Это значение вычисляется по алгоритму CRC-32.

## 12.3. Доступ к среде и передача данных

Метод доступа, используемый в сетях Ethernet на разделяемой проводной среде, называется CSMA/CD (Carrier Sense Multiple Access with Collision Detection — прослушивание несущей частоты с множественным доступом и распознаванием коллизий).

Все компьютеры в сети на разделяемой среде имеют возможность немедленно (с учетом задержки распространения сигнала в физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду. Говорят, что среда, к которой подключены все станции, работает в режиме коллективного доступа (Multiply Access, MA).

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая еще называется несущей частотой (Carrier Sense, CS).

Признаком незанятости среды является отсутствие на ней несущей частоты.

Если среда свободна, то узел имеет право начать передачу кадра. В примере, показанном на рис. 22, узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В коаксиальном кабеле

сигналы передатчика узла 1 распространяются в обе стороны, так что их получают все узлы сети. Кадр данных всегда сопровождается преамбулой, которая состоит из 7 байт, каждый из которых имеет значение 10101010, и 8-го байта, равного 10101011. Последний байт носит название ограничителя начала кадра. Преамбула нужна для вхождения приемника в побитовую и побайтовую синхронизацию с передатчиком. Наличие двух единиц, идущих подряд, говорит приемнику о том, что преамбула закончилась и следующий бит является началом кадра.

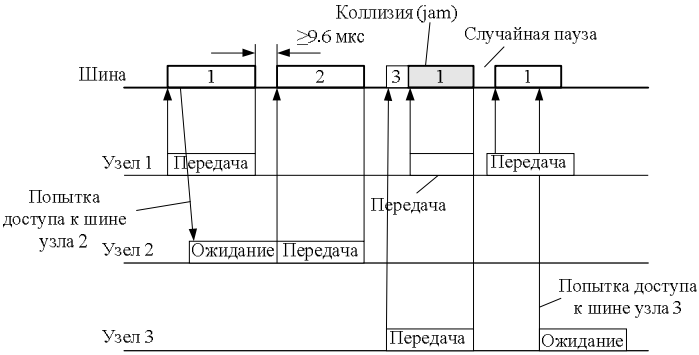



**Рис. 22** Метод случайного доступа CSMA/CD

Все станции, подключенные к кабелю, начинают записывать байты передаваемого кадра в свои внутренние буферы. Первые 6 байт кадра содержат адрес назначения. Та станция, которая узнает собственный адрес в заголовке кадра, продолжает записывать его содержимое в свой внутренний буфер, а остальные станции на этом прием кадра прекращают. Станция назначения обрабатывает полученные данные и передает их вверх по своему стеку. Кадр Ethernet содержит не только адрес назначения, но и адрес источника данных, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаруживает, что среда занята — на ней присутствует несущая частота, — поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу, равную межпакетному интервалу (Inter Packet Gap, IPG) в 9,6 мкс. Эта пауза нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания

технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра.

## 12.4. Спецификации физической среды

При стандартизации технологии Ethernet рабочей группой IEEE 802.3 вариант Ethernet на «толстом» коаксиальном кабеле получил название 10Base-5.

Число 10 в этом названии обозначает номинальную битовую скорость передачи данных стандарта, то есть 10 Мбит/с, а слово «Base» — метод передачи на одной базовой частоте (в данном случае 10 МГц). Последний символ в названии стандарта физического уровня обозначает тип кабеля, в данном случае 5 отражает тот факт, что диаметр «толстого» коаксиала равен 0,5 дюйма. Данная система обозначения типа физического уровня Ethernet сохранилась до настоящего времени.

Наиболее популярными спецификациями физической среды Ethernet, для скорости передачи данных 10 Мбит/с являются следующие:

**1.** 10Base-5 — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом, максимальную длину сегмента – 500 м (без повторителей), максимальное количество узлов подключаемых к сегменту — 100, максимальное число сегментов — 5 (4 повторителя), из которых только 3 могут использоваться для подключения узлов, а 2 играют роль удлинителей сети;

**2.** 10Base-2 — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом, максимальную длину сегмента — 185 м (без повторителей), максимальное количество узлов подключаемых к сегменту — 30, максимальное число сегментов — 5 (4 повторителя), из которых только 3 могут использоваться для подключения узлов, а 2 играют роль удлинителей сети;

**3.** 10Base-T — кабель на основе неэкранированной витой пары (UTP). Образует звездообразную топологию на основе концентратора (многопортового повторителя). Расстояние между концентратором и конечным узлом — не более 100 м. Между любыми двумя узлами сети может быть не более 4-х концентраторов (так называемое «правило 4-х хабов»);

**4.** 10Base-F — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T, но расстояние между концентратором и конечным узлом может достигать 2000 м. Правило 4-х хабов остается в силе.

Одним из существенных недостатков Ethernet на коаксиальном кабеле являлось отсутствие оперативной информации о состоянии кабеля и сложность нахождения места его повреждения. Поэтому поиск неисправностей стал привычной процедурой и головной болью многочисленной армии сетевых администраторов коаксиальных сетей Ethernet.

К этому времени телефонные компании уже достаточно давно применяли многопарный кабель на основе неэкранированной витой пары для подключения телефонных аппаратов внутри зданий. Идея приспособить этот популярный вид кабеля для локальных сетей оказалась очень плодотворной, так как многие здания уже были оснащены нужной кабельной системой.

Физическая структуризация сетей, построенных на основе витой пары, повышает надежность и упрощает обслуживание сети, поскольку в этом случае появляется возможность контролировать состояние и устранять отказы отдельных кабельных отрезков, подключающих конечные узлы к концентраторам. В случае обрыва, короткого замыкания или неисправности сетевого адаптера работа сети может быть быстро восстановлена путем отключения соответствующего сегмента кабеля.

Для контроля целостности физического соединения между двумя непосредственно соединенными портами в стандарте 10Base-T введен так называемый тест целостности соединения (Link Integrity Test, LIT). Эта процедура заключается в том, что в те периоды, когда порт не посылает или получает кадры данных, он посылает своему соседу импульсы длительностью 100 нс через каждые 16 мс. Если порт принимает такие импульсы от своего соседа, то он считает соединение работоспособным и, как правило, индицирует это зеленым светом светодиода.

Независимо от используемого физического уровня в стандартах Ethernet на 10 Мбит/с вводится ограничение на максимальное количество узлов, подключаемых к разделяемой среде. Это ограничение составляет 1024 узла.

## 13. Технологии Token Ring и FDDI

Token Ring и FDDI это функционально намного более сложные технологии, чем Ethernet. Механизм доступа к среде в сетях Token

Ring и FDDI является более детерминированными, чем в сетях Ethernet.

Рассмотрим его на примере сети Token Ring (рис. 23). В сети станции связаны в кольцо так, что любая станция непосредственно получает данные только от одной станции – той, которая является предыдущей в кольце, а передает данные своему ближайшему соседу вниз по потоку данных. Скорость передачи данных в первых сетях Token Ring, разработанных компанией IBM, была всего 4 Мбит/с, но затем была повышена до 16 Мбит/с. Основная среда передачи данных — витая пара. Для адресации станций сети Token Ring (и FDDI) используют MAC-адреса того же формата, что и Ethernet.

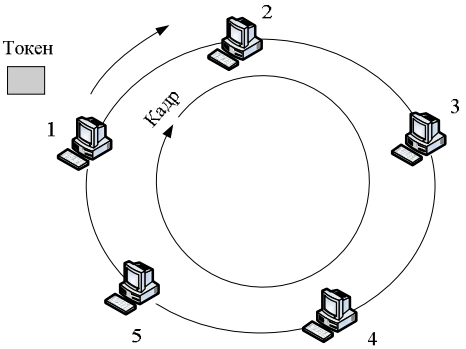



**Рис. 23** Сеть Token Ring

Метод доступа Token Ring основан на передаче от узла к узлу специального кадра — токена, или маркера доступа, при этом только узел, владеющий токеном, может передавать свои кадры в кольцо, которое становится в этом случае разделяемой средой.

Существует ограничение на период монопольного использования среды — это так называемое время удержания токена, по истечении которого станция обязана передать токен своему соседу по кольцу. Кроме того, такие ситуации, как неопределенное время ожидания доступа к среде характерные для Ethernet, здесь исключены (по крайней мере, в тех случаях, когда сетевые адаптеры станций исправны и работают без сбоев). Максимальное время ожидания равно произведению времени удержания токена на количество станций в кольце. Так как станция, получившая токен, но не имеющая в тот момент кадров для передачи, передает токен следующей станции, то время ожидания может быть меньше.

Отказоустойчивость сети Token Ring определяется использованием в сети повторителей для создания кольца. Каждый

такой повторитель имеет несколько портов, которые образуют кольцо за счет внутренних связей между передатчиками и приемниками. В случае отказа или отсоединения станции повторитель организует обход порта этой станции, так что связность кольца не нарушается. Поддержка чувствительного к задержкам трафика достигается за счет системности приоритетов кадров. Решение о приоритете конкретного кадра принимает передающая станция. Токен также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей токен только в том случае, если приоритет кадра, который она хочет передать, выше приоритета токена (или равен ему). В противном случае станция обязана передать токен следующей по кольцу станции. Благодаря более высокой, чем в сетях Ethernet, скорости, детерминированности распределения пропускной способности сети между узлами, лучших эксплуатационных характеристик (обнаружение и изоляция неисправностей) сети Token Ring были предпочтительным выбором для таких чувствительных к подобным показателям приложений, как банковские системы и системы управления предприятием.

Технологию FDDI можно считать усовершенствованным вариантом Token Ring, так как в ней, как и в Token Ring, используется метод доступа к среде, основанный на передаче токена, а также кольцевая топология связей, но вместе с тем FDDI работает на более высокой скорости и имеет более совершенный механизм отказоустойчивости. Технология FDDI стала первой технологией локальных сетей, в которой оптическое волокно, начавшее применяться в телекоммуникационных сетях с 70-х годов прошлого века, было использовано в качестве разделяемой среды передачи данных. За счет применения оптических систем скорость передачи данных удалось повысить до 100 Мбит/с (позже появилось оборудование FDDI на витой паре, работающее на той же скорости).

В тех случаях, когда нужно было обеспечить высокую надежность сети FDDI, применялось двойное кольцо (рис. 24). В нормальном режиме станции используют для передачи данных и токена доступа первичное кольцо, а вторичное простаивает. В случае отказа, например, при обрыве кабеля между станциями 1 и 2, как показано на рис. 24, первичное кольцо объединяется с вторичным, вновь образуя единое кольцо. Этот режим работы сети называется режимом свертывания колец. Операция свертывания производится средствами повторителей (не показанных на рисунке) и/или сетевых адаптеров FDDI. Для упрощения этой процедуры, данные по первичному кольцу всегда передаются в одном направлении (на

диаграммах это направление изображается против часовой стрелки), по вторичному — в обратном (изображается по часовой стрелке). Поэтому образование общего кольца из двух колец, передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, позволяет правильно передавать и принимать информацию соседними станциями.

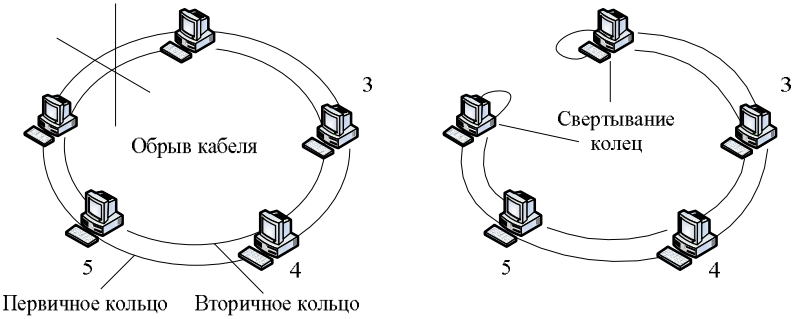



**Рис. 24** Отказоустойчивость в сети FDDI

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить факт наличия отказа в сети, а затем произвести необходимое реконфигурирование. Технология FDDI расширяет механизмы обнаружения отказов технологии Token Ring за счет резервных связей, которые предоставляет второе кольцо.

# 14. Беспроводные локальные сети IEEE 802.11(WLAN)

## 14.1. Проблемы и области применения беспроводных локальных сетей

Беспроводные локальные сети (Wireless Local Area Network, WLAN) в некоторых случаях являются предпочтительным по сравнению с проводной сетью решением, а иногда просто единственно возможным. В WLAN сигнал распространяется с помощью электромагнитных волн высокой частоты.

Преимущество беспроводных локальных сетей очевидно — их проще и дешевле разворачивать и модифицировать, так как вся громоздкая кабельная инфраструктура оказывается излишней. Еще одно преимущество — обеспечение мобильности пользователей. Однако основной проблемой является неустойчивая и непредсказуемая беспроводная среда (например, помехи от

разнообразных бытовых приборов и других телекоммуникационных систем, атмосферные помехи и отражения сигнала).

Локальные сети — это, прежде всего, сети зданий, а распространение радиосигнала внутри здания еще сложнее, чем вне него.

Методы расширения спектра помогают снизить влияние помех на полезный сигнал, кроме того, в беспроводных сетях широко используются прямая коррекция ошибок (FEC) и протоколы с повторной передачей потерянных кадров.

Неравномерное распределение интенсивности сигнала приводит не только к битовым ошибкам передаваемой информации, но и к неопределенности зоны покрытия беспроводной локальной сети. В проводных локальных сетях такой проблемы нет. Беспроводная локальная сеть не имеет точной области покрытия. В действительности, сигнал может быть настолько ослаблен, что устройства, находящиеся в предполагаемых пределах зоны покрытия, вообще не могут принимать и передавать информацию.

На рис. 25(а) показана фрагментированная локальная сеть. Неполносвязность беспроводной сети порождает проблему доступа к разделяемой среде, известную под названием скрытого терминала. Проблема возникает в том случае, когда два узла находятся вне зон досягаемости друг друга (узлы *A* и *C* на рис. 25(а)), и существует третий узел *B*, который принимает сигналы как от *A*, так и от *C*. Предположим, что в радиосети используется традиционный метод доступа, основанный на прослушивании несущей, например CSMA/CD. В данном случае коллизии будут возникать значительно чаще, чем в проводных сетях. Пусть, например, узел *B* занят обменом с узлом *A*. Узлу *C* сложно определить, что среда занята, он может посчитать ее свободной и начать передавать свой кадр. В результате сигналы в районе узла *B* исказятся, то есть произойдет коллизия, вероятность возникновения которой в проводной сети была бы неизмеримо ниже.

Распознавание коллизий затруднено в радиосети еще и потому, что сигнал собственного передатчика существенно подавляет сигнал удаленного передатчика, и распознать искажение сигнала чаще всего невозможно.

В методах доступа применяемых в беспроводных сетях, отказываются не только от прослушивания несущей, но и распознания коллизий. Вместо этого в них используют методы предотвращения коллизий, включая методы опроса.

Применение базовой станции может улучшить связность сети

(рис. 25(б)). Базовая станция обычно обладает большей мощностью, а ее антенна устанавливается так, чтобы более равномерно и беспрепятственно покрывать нужную территорию. В результате все узлы беспроводной локальной сети получают возможность обмениваться данными с базовой станцией, которая транзитом передает данные между узлами.

Беспроводные локальные сети считаются перспективными для таких применений, в которых сложно или невозможно использовать проводные сети.

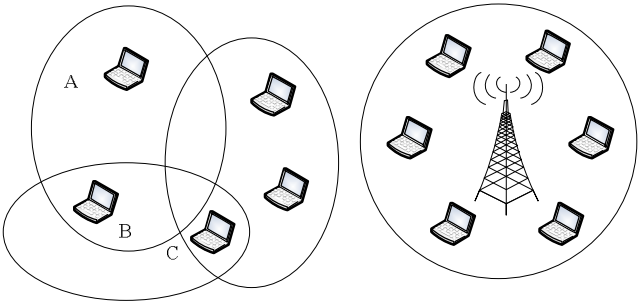



**Рис. 25** Связность беспроводной локальной сети:
а — специализированная беспроводная сеть, б — беспроводная сеть с базовой станцией

Области применения беспроводных локальных сетей:

**1.** Домашние локальные сети. Когда в доме появляется несколько компьютеров, организация домашней локальной сети становится актуальной проблемой.

**2.** Резидентный доступ альтернативных операторов связи, у которых нет проводного, доступа к клиентам, проживающим в многоквартирных домах,

**3.** Так называемый «кочевой» доступ в аэропортах, железнодорожных вокзалах и т. п.

**4.** Организация локальных сетей в зданиях, где нет возможности установить современную кабельную систему, например в исторических зданиях с оригинальным интерьером.

**5.** Организация временных локальных сетей, например, при проведении конференций.

**6.** Расширения локальных сетей. Иногда одно здание предприятия, например испытательная лаборатория или цех, может быть расположено отдельно от других. Небольшое число рабочих мест в таком здании делает крайне невыгодным прокладку к нему

отдельного кабеля, поэтому беспроводная связь оказывается более рациональным вариантом.

**7.** Мобильные локальные сети. Если пользователь хочет получать услуги сети, перемещаясь из помещения в помещение или из здания в здание, то здесь конкурентов у беспроводной локальной сети просто нет. Классическим примером такого пользователя является врач, совершающий обход и пользующийся своим ноутбуком для связи с базой данных больницы.

## 14.2. Топологии локальных сетей стандарта 802.11

Стандарт 802.11 поддерживает два типа топологий локальных сетей: с базовым и с расширенным наборами услуг.

Сеть с базовым набором услуг (Basic Service Set, BSS) образуется отдельными станциями. Базовая станция отсутствует. Узлы взаимодействуют друг с другом непосредственно (рис. 26). Для того чтобы войти в сеть BSS, станция должна выполнить процедуру присоединения.

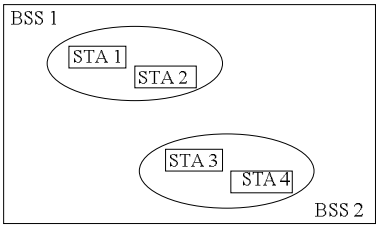



**Рис. 26** Сети с базовым набором услуг

Сети BSS могут находиться друг от друга на значительном расстоянии, могут частично или полностью перекрываться.

Станции могут использовать разделяемую среду для того, чтобы передавать данные:

- непосредственно друг другу в пределах одной сети BSS;
- в пределах одной сети BSS транзитом через точку доступа;
- между разными сетями BSS через две точки доступа и распределенную систему;
- между сетью BSS и проводной локальной сетью через точку доступа, распределенную систему и портал.

В сетях, обладающих инфраструктурой, некоторые станции сети являются базовыми, или, в терминологии стандарта 802.11 - точками доступа (Access Point, AP). Станция, которая выполняет функции AP, является членом какой-нибудь сети BSS (рис. 27).

Все базовые станции сети связаны между собой с помощью распределенной системы (Distribution System, DS), в качестве которой может использоваться та же среда (то есть радио- или инфракрасные волны), что и среда взаимодействия между станциями, или же отличная от нее, например проводная.

Точки доступа вместе с распределенной системой поддерживают службу распределенной системы (Distribution System Service, DSS). Задачей DSS является передача пакетов между станциями, которые по каким-то причинам не могут или не хотят взаимодействовать между собой непосредственно. Наиболее очевидной причиной использования DSS является принадлежность станций разным сетям BSS. В этом случае они передают кадр своей точке доступа, которая через DS передает его точке доступа, обслуживающей сеть BSS со станцией назначения.

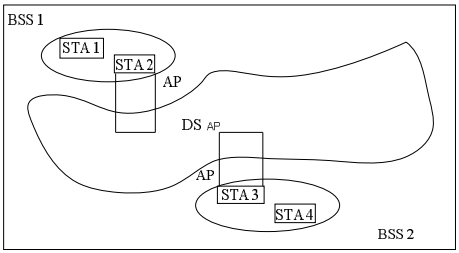



**Рис. 27** Сеть с расширенным набором услуг

Сеть с расширенным набором услуг (Extended Service Set, ESS) состоит из нескольких сетей BSS, объединенных распределенной средой.

Сеть ESS обеспечивает станциям мобильность — они могут переходить из одной сети BSS в другую. Эти перемещения обеспечиваются функциями уровня MAC рабочих и базовых станций, потому они совершенно прозрачны для уровня LLC. Сеть ESS может также взаимодействовать с проводной локальной сетью. Для этого в распределенной системе должен присутствовать портал.

# 15. Мост как предшественник и функциональный аналог коммутатора

## 15.1. Логическая структуризация сетей и мосты

Мост локальной сети (LAN bridge), или просто мост, появился как средство построения крупных локальных сетей на разделяемой среде,

так как невозможно построить достаточно крупную сеть на одной разделяемой среде

Использование единой разделяемой среды в сети Ethernet приводит к нескольким очень жестким ограничениям:

- общий диаметр сети не может быть больше 2500 м;
- количество узлов не может превышать 1024.

На рис. 28 показана зависимость задержки доступа к среде передачи от загруженности сети, полученные для сетей Ethernet, Token Ring и FDDI путем имитационного моделирования.

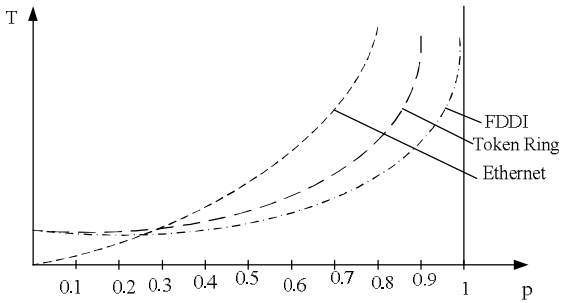



**Рис. 28** Зависимость задержки доступа к среде передачи данных для технологий Ethernet, Token Ring и FDDI

Как видно из рисунка, всем технологиям присуща одинаковая картина экспоненциального роста величины задержек доступа при увеличении коэффициента использования сети. Однако их отличает порог, при котором наступает резкий перелом в поведении сети, когда почти прямолинейная зависимость переходит в экспоненциальную. Для всего семейства технологий Ethernet — это 30-50 % (сказывается эффект коллизий), для технологии Token Ring — 60 %, а для технологии FDDI — 70-80 %.

Ограничения, возникающие из-за использования единой разделяемой среды, можно преодолеть. Для этого необходимо выполнить логическую структуризацию сети, то есть сегментировать единую разделяемую среду на несколько и соединить полученные сегменты сети коммуникационным устройством, которое не передает данные побитно, как повторитель, а базирует кадры и передает их затем в тот или иной сегмент в зависимости от адреса назначения кадра (рис. 29).

Нужно отличать логическую структуризацию от физической. Концентраторы стандарта 10Base-T позволяют построить сеть, состоящую из нескольких сегментов кабеля на витой паре, но это —

физическая структуризация, так как логически все эти сегменты представляют собой единую разделяемую среду.

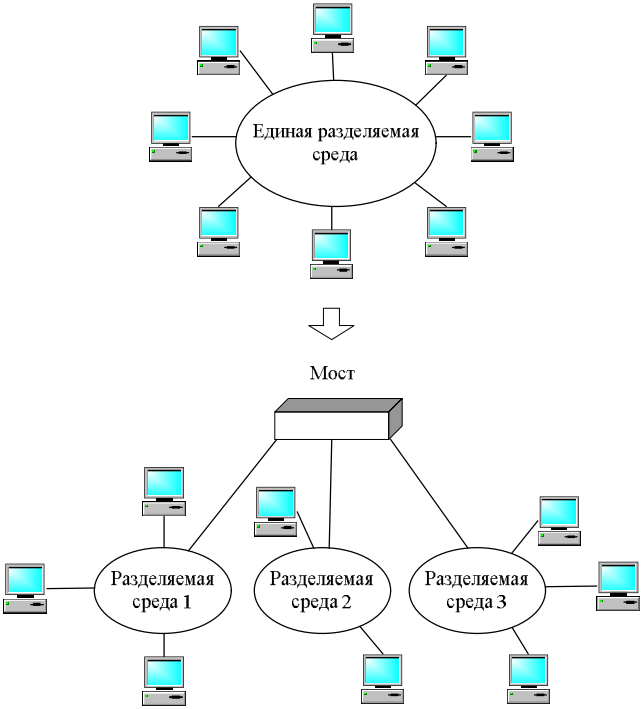



**Рис. 29** Логическая структуризация сети

Мост долгое время использовался в качестве логического устройства для структуризации локальных сетей. В настоящее время мосты заменили коммутаторы. Их алгоритм работы повторяет алгоритм работы моста. Отличием служит только гораздо более высокая производительность коммутаторов.

Помимо мостов/коммутаторов для структуризации локальных сетей можно использовать маршрутизаторы, но они являются более сложными и дорогими устройствами, всегда требующими ручного конфигурирования, поэтому их применение в локальных сетях ограничено.

Логическая структуризация локальной сети позволяет решить несколько задач, основные из которых — это повышение производительности, гибкости и безопасности, а также улучшение управляемости сети.

При построении сети как совокупности сегментов каждый из них может быть адаптирован к специфическим потребностям рабочей группы или отдела. Это означает повышение гибкости сети. Процесс разбиения сети на логические сегменты можно рассматривать и в обратном направлении, как процесс создания большой сети из уже имеющихся небольших сетей.

Устанавливая различные логические фильтры на мостах/коммутаторах, можно контролировать доступ пользователей к ресурсам других сегментов, чего не позволяют делать повторители. Так достигается повышение безопасности данных.

Побочным эффектом снижения трафика и повышения безопасности данных является упрощение управления сетью, то есть улучшение её управляемости. Проблемы очень часто локализуются внутри сегмента. Сегменты образуют логические домены управления сетью.

## 15.2. Алгоритм прозрачного моста IEEE 802.1D

В локальных сетях 80-х и 90-х годов применялись мосты нескольких типов:

- прозрачные мосты (для технологии Ethernet);
- мосты с маршрутизацией от источника (для технологии Token Ring);
- транслирующие мосты (для соединения технологий Ethernet и Token Ring).

Слово «прозрачный» в названии алгоритма отражает тот факт, что мосты и коммутаторы в своей работе не учитывают существование в сети сетевых адаптеров конечных узлов, концентраторов и повторителей. В то же время и перечисленные сетевые устройства функционируют, «не замечая» присутствия в сети мостов и коммутаторов.

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

Рассмотрим процесс автоматического создания таблицы продвижения моста и ее использования на примере простой сети, представленной на рис. 30:

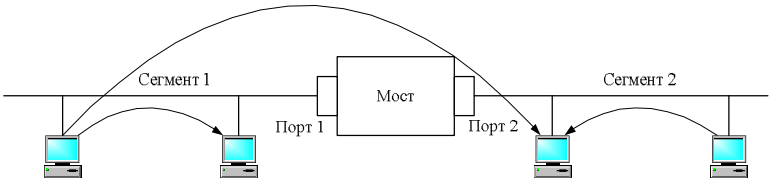
Мост соединяет два сетевых сегмента. Сегмент *1* составляют

компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту *1* моста, а сегмент *2* - компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту *2* моста. В исходном состоянии мост не знает о том, компьютеры с какими MAC-адресами подключены к каждому из его портов. В этой ситуации мост просто передает любой захваченный и буферизованный кадр на все свои порты за исключением того порта, от которого этот кадр получен. В примере у моста только два порта, поэтому он передает кадры с порта *1* на порт *2*, и наоборот. Отличие работы моста в этом режиме от повторителя заключается в том, что он передает кадр, предварительно буферизуя его, а не бит за битом, как это делает повторитель. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды.

Одновременно с передачей кадра на все порты мост изучает адрес источника кадра и делает запись о его принадлежности к тому или иному сегменту в своей адресной таблице. Эту таблицу также называют таблицей фильтрации, или продвижения. Например, получив на порт *1* кадр от компьютера *1*, мост делает первую запись в своей адресной таблице:

MAC-адрес 1 — порт 1.

Эта запись означает, что компьютер, имеющий MAC-адрес *1*, принадлежит сегменту, подключенному к порту *1* коммутатора. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро мост построит полную адресную таблицу сети, состоящую из 4-х записей — по одной записи на узел (рис. 30).




| MAC-адрес | Порт |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |

**Рис. 30** Принцип работы прозрачного моста/коммутатора

При каждом поступлении кадра на порт моста он, прежде всего, пытается найти адрес назначения кадра в адресной таблице. Продолжим рассмотрение действий моста на примере рис. 30:

**1.** При получении кадра, направленного от компьютера *1*

компьютеру *3*, мост просматривает адресную таблицу на предмет совпадения адреса в какой-либо из её записей с адресом назначения— MAC-адресом узла *3*. Запись с искомым адресом имеется в адресной таблице.

**2.** Мост выполняет второй этап анализа таблицы — проверяет, находятся ли компьютеры с адресами источника и назначения в одном сегменте. В примере компьютер *1* (MAC-адрес *1*) и компьютер *3* (MAC-адрес *3*) находятся в разных сегментах. Следовательно, мост выполняет операцию продвижения (forwarding) кадра — передает кадр на порт *2*, ведущий в сегмент получателя, получает доступ к сегменту и передает туда кадр.

**3.** Если бы оказалось, что компьютеры принадлежали бы одному и тому же сегменту, то кадр просто был бы удален из буфера. Такая операция называется фильтрацией (fikering).

**4.** Если бы запись о MAC-адресе узла *3* отсутствовала в адресной таблице, то есть адрес узла назначения не был известен мосту, то он бы передал кадр на все свои порты, кроме порта узла отправителя кадра, как и на начальной стадии процесса обучения.

Процесс обучения моста никогда не заканчивается и происходит одновременно с продвижением и фильтрацией кадров. Мост постоянно следит за адресами источника буферизуемых кадров, чтобы автоматически приспосабливаться к изменениям, происходящим в сети: перемещениям компьютеров из одного сегмента сети в другой, отключению и появлению новых компьютеров.

Входы адресной таблицы могут быть динамическими, создаваемыми в процессе самообучения моста, и статическими, создаваемыми вручную администратором сети. Статические записи, не имеют срока жизни, что дает администратору возможность влиять на работу моста, например, ограничивать передачу данных от узлов с определёнными адресами из одного сегмента в другой.

Динамические записи имеют срок жизни. При создании или обновлении записи в адресной таблице с ней связывается отметка времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время мост не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность мосту автоматически реагировать на перемещения компьютера из сегмента в сегмент. При отключении компьютера от старого сегмента запись о нём автоматически удаляется из адресной таблицы. После подключения компьютера к другому сегменту его кадры начнут попадать в буфер моста через другой порт, и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Кадры с широковещательными MAC-адресами, как и кадры с неизвестными адресами назначения, передаются мостом на все его порты. Такой режим распространения кадров называется затоплением сети (flooding). Наличие мостов в сети не препятствует распространению широковещательных кадров по всем сегментам сети. Однако это является достоинством только тогда, когда широковещательный адрес выработан корректно работающим узлом.

Нередко в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сетевой адаптер начинает работать некорректно, а именно постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом. Мост в соответствии со своим алгоритмом передает ошибочный трафик во все сегменты. Такая ситуация называется широковещательным штормом (broadcast, storm).

На рис. 31 показана типичная структура моста. Функции доступа к среде при приеме и передаче кадров выполняют микросхемы MAC, которые идентичны микросхемам сетевого адаптера.

Протокол, реализующий алгоритм коммутатора, располагается между уровнями MAC HLLC.

На рис. 32 показана копия экрана терминала с адресной таблицей моста.

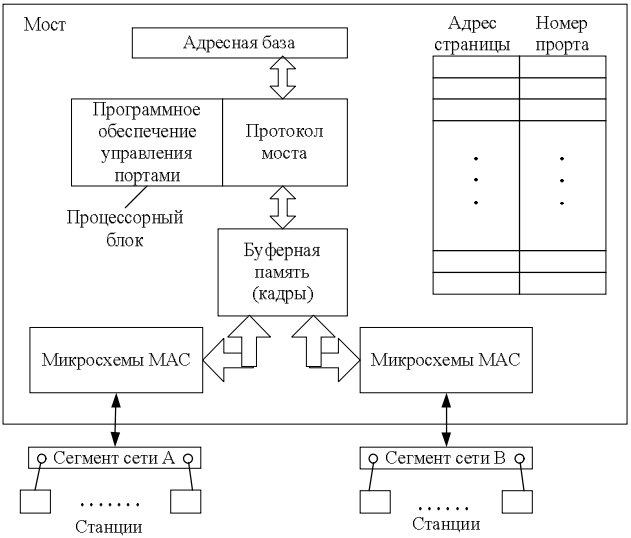



**Рис. 31** Структура моста/коммутатора

Из выводимой на экран адресной таблицы видно, что сеть состоит из двух сегментов — LAN A и LAN B. В сегменте LAN A имеются, по крайней мере, 3 станции, а в сегменте LAN B — 2 станции. Четыре адреса, помеченные звездочками, являются статическими, то есть назначенными администратором вручную. Адрес, помеченный плюсом, является динамическим адресом с истекшим сроком жизни.

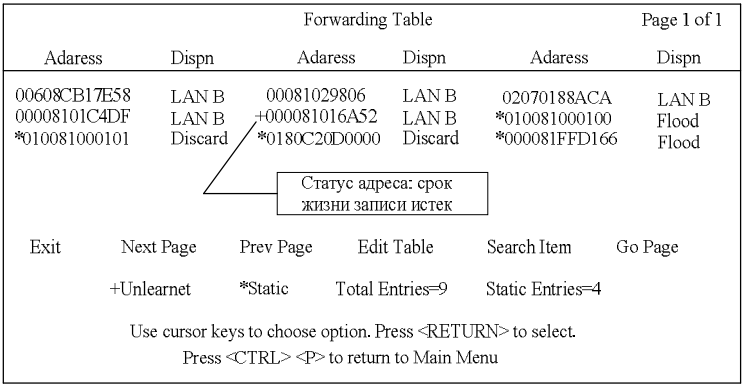



**Рис. 32** Адресная таблица коммутатора

Таблица имеет поле Dispn — «disposition» (это «распоряжение» мосту о том, какую операцию нужно проделать с кадром, имеющим данный адрес назначения). Обычно при автоматическом составлении таблицы в этом поле ставится условное обозначение порта назначения, но при ручном задании адреса в это поле можно внести нестандартную операцию обработки кадра. Например, операция Flood (затопление) заставляет мост распространять кадр в широковещательном режиме, несмотря на то, что его адрес назначения не является широковещательным. Операция Discard (отбросить) говорит мосту, что кадр с таким адресом не нужно передавать на порт назначения. Вообще говоря, операции, задаваемые в поле Dispn, определяют особые условия фильтрации кадров, дополняющие стандартные условия их распространения. Такие условия обычно называют пользовательскими фильтрами.

## 15.3. Топологические ограничения при применении мостов в локальных сетях

Рассмотрим это ограничение на примере сети, показанной на рис. 33.

Два сегмента Ethernet параллельно соединены двумя мостами, так чтобы образовалась петля. Пусть новая станция с MAC-адресом 123 впервые начинает работу в данной сети. Обычно начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых станция заявляет о своем существовании и одновременно ищет серверы сети.

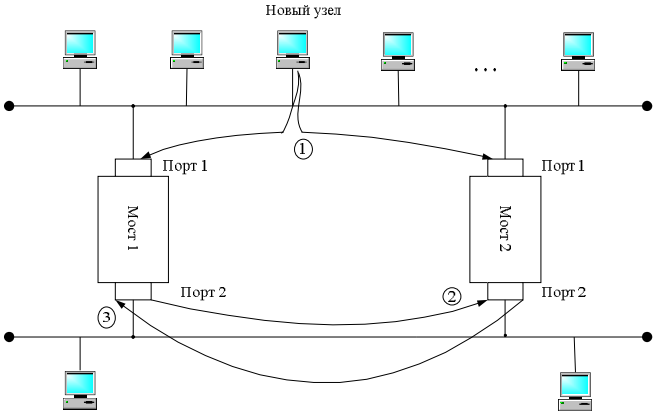



**Рис. 33** Влияние замкнутых маршрутов на работу коммутаторов

На этапе 1 станция посылает первый кадр с широковещательным адресом назначения и адресом источника 123 в свой сегмент. Кадр попадает как в мост 1, так и в мост 2. В обоих мостах новый адрес источника 123 заносится в адресную таблицу с пометкой о его принадлежности сегменту 1, то есть создается новая запись вида:

MAC-адрес 123 - Порт 1.

Так как адрес назначения широковещательный, то каждый мост должен передать кадр на сегмент 2. Эта передача происходит поочередно в соответствии с методом случайного доступа технологии Ethernet. Пусть первым доступ к сегменту 2 получает мост 1 (этап 2 на рис. 33). При появлении кадра на сегменте 2 мост 2 принимает его в свой буфер и обрабатывает. Он видит, что адрес 123 уже есть в его адресной таблице, но пришедший кадр является более свежим, и он решает, что адрес 123 принадлежит сегменту 2, а не 1. Поэтому мост 2 корректирует содержимое базы и делает запись о том, что адрес 123 принадлежит сегменту 2:

MAC-адрес 123 - Порт 2.

Аналогично поступает мост 1, когда мост 2 передает свою копию кадра на сегмент 2. Далее перечислены последствия наличия петли в

сети:

- «Размножение» кадра, то есть появление нескольких его копий (в данном случае — двух, но если бы сегменты были соединены тремя мостами — то трех и т. д.).
- Бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком.
- Постоянная перестройка мостами своих адресных таблиц, так как кадр с адресом источника 123 будет появляться то на одном порту, то на другом.

В целях исключения всех этих нежелательных эффектов мосты/коммутаторы нужно применять так, чтобы между логическими сегментами не было петель, то есть строить с помощью коммутаторов только древовидные структуры, гарантирующие наличие единственного пути между любыми двумя сегментами. Тогда кадры от каждой станции будут поступать на мост/коммутатор всегда с одного и того же порта, и коммутатор сможет правильно решать задачу выбора рационального маршрута в сети.

Возможна и другая причина возникновения петель. Так, для повышения надежности желательно иметь между мостами/коммутаторами резервные связи, которые не участвуют в нормальной работе основных связей по передаче информационных кадров станций, но при отказе какой-либо основной связи образуют новую связную рабочую конфигурацию без петель.

Избыточные связи необходимо блокировать, то есть переводить их в неактивное состояние. В сетях с простой топологией эта задача решается вручную путем блокирования соответствующих портов мостов/коммутаторов. В больших сетях со сложными связями используются алгоритмы, которые позволяют решать задачу обнаружения петель автоматически.

## 16. Коммутаторы. Параллельная коммутация

В начале 90-х годов с появлением быстрых протоколов появились коммутаторы. Они предназначались для обслуживания потока, поступающего на каждый порт. В устройство ставился отдельный специализированный процессор, который реализовывал алгоритм прозрачного моста. По сути, коммутатор — это мультипроцессорный мост, способный параллельно продвигать кадры сразу между всеми парами своих портов.

Производительность коммутаторов на несколько порядков выше,

чем мостов – коммутаторы могут передавать до нескольких десятков, а иногда и сотен миллионов кадров в секунду, в то время как мосты обычно обрабатывали 3 – 5 тысяч кадров в секунду.

Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana, представлена на рис. 34.

Каждый из 8 портов 10Base-T обслуживается одним процессором пакетов Ethernet (Ethernet Packet Processor, EPP). Кроме того, коммутатор имеет системный модуль, который координирует работу всех процессоров EPP, в частности ведет общую адресную таблицу коммутатора для передачи кадров между портами - используется коммутационная матрица. Она функционирует по принципу коммутации каналов, соединяя порты коммутатора.


**Рис. 34** Структура коммутатора EtherSwitch компании Kalpana

При поступлении кадра в какой-либо порт соответствующий процессор EPP буферизует несколько первых байтов кадра, чтобы прочитать адрес назначения. После получения адреса назначения процессор сразу же приступает к обработке кадра, не дожидаясь прихода остальных его байтов:

**1.** Процессор EPP просматривает свой кэш адресной таблицы, и если не находит такого нужного адреса, обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров EPP. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку, которую тот буферизует в своем кэше для последующего использования.

**2.** Если адрес назначения найден в адресной таблице, и кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра.

**3.** Если же адрес найден, и кадр нужно передать на другой порт, процессор, продолжая прием кадра в буфер, обращается к коммутационной матрице, пытаясь установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу назначения. Коммутационная матрица способна помочь только в том случае, если порт адреса назначения в этот момент свободен, то есть, не соединен с другим портом данного коммутатора.

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

**5.** После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet по алгоритму CSMA/CD, байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байтов принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра (рис. 35).

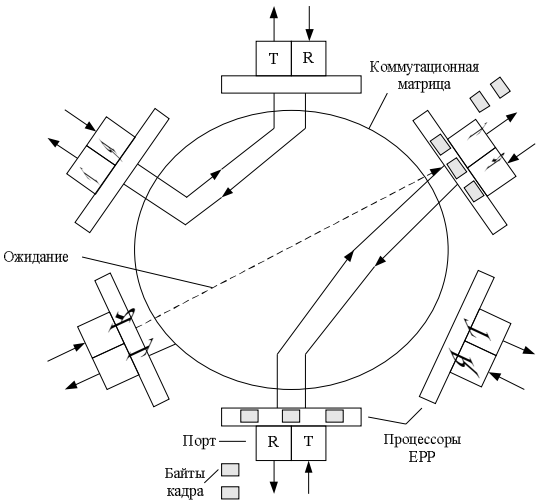



**Рис. 35** Передача кадра через коммутационную матрицу

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

**1.** Прием первых байтов кадра процессором входного порта, включая прием байтов адреса назначения.

**2.** Поиск адреса назначения в адресной таблице коммутатора (в кэше процессора или в общей таблице системного модуля).

**3.** Коммутация матрицы.

**4.** Прием остальных байтов кадра процессором входного порта.

**5.** Прием байтов кадра (включая первые) процессором выходного порта через коммутационную матрицу.

**6.** Получение доступа к среде процессором выходного порта

**7.** Передача байтов кадра процессором выходного порта в сеть.

## 16.1. Дуплексный режим работы коммутатора

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

Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров.

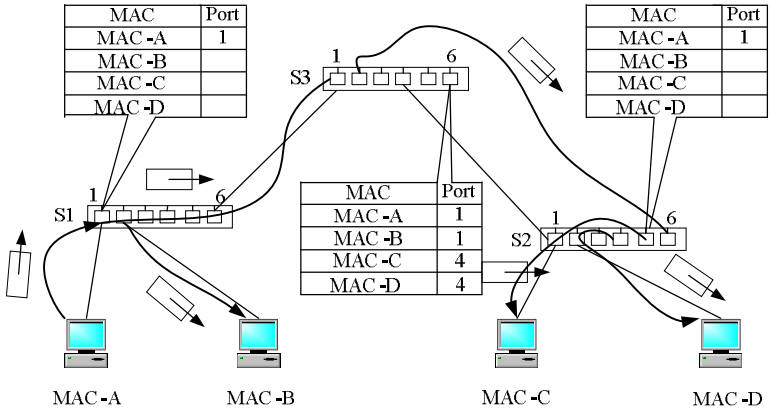



**Рис. 36** Полностью коммутируемая сеть Ethernet

В дуплексном режиме одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для отдельных дуплексных каналов передачи данных, и он всегда использовался в протоколах глобальных сетей. При дуплексной связи порты Ethernet стандарта 10 Мбит/с могут передавать данные со скоростью 20 Мбит/с, но 10 Мбит/с в каждом направлении.

Постепенно коммутаторы стали вытеснять концентраторы, так как цены коммутаторов постоянно снижались, а их производительность росла. Производительность увеличивалась за счет поддержки не только технологии Ethernet со скоростью 10 Мбит/с, но и всех последующих более скоростных версий этой технологии (Fast Ethernet со скоростью 100 Мбит/с, Gigabit Ethernet со скоростью 1 Гбит/с и 10G Ethernet со скоростью 10 Гбит/с). Этот процесс завершился вытеснением концентраторов Ethernet и переходом к полностью коммутируемым сетям, пример такой сети показан на рис. 36:

В полностью коммутируемой сети Ethernet все порты работают в дуплексном режиме, а продвижение кадров осуществляется на основе MAC-адресов.

## 17. Виртуальные локальные сети

Виртуальной локальной сетью (Virtual Local Area Network, VLAN) называется группа узлов сети, трафик которой, в том числе широковещательный, на канальном уровне полностью изолирован от трафика других узлов сети.

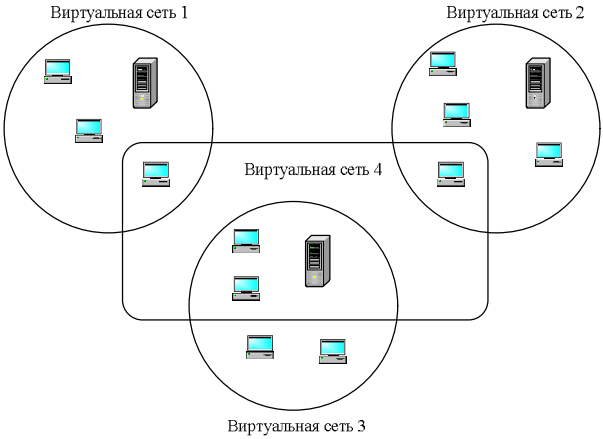



**Рис. 37** Виртуальные локальные сети

Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна независимо от типа адреса (уникального, группового или широковещательного). В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра.

Виртуальные локальные сети могут перекрываться, если один или несколько компьютеров входят в состав более чем одной виртуальной сети. На рис. 37 сервер электронной почты входит в состав виртуальных сетей 3 и 4. Это означает, что его кадры передаются коммутаторами всем компьютерам, входящим в эти сети. Если же какой-то компьютер входит в состав только виртуальной сети 3, то его кадры до сети 4 доходить не будут, но он может взаимодействовать с компьютерами сети 4 через общий почтовый сервер. Такая схема защищает виртуальные сети друг от друга не полностью, например, широковещательный шторм, возникший на сервере электронной почты, затопит и сеть 3, и сеть 4.

Говорят, что виртуальная сеть образует домен широковещательного трафика по аналогии с доменом коллизий, который образуется повторителями сетей Ethernet.

## 17.1. Назначение виртуальных сетей

Как мы видели на примере из предыдущего раздела, с помощью пользовательских фильтров можно вмешиваться в нормальную работу коммутаторов и ограничивать взаимодействие узлов локальной сети в соответствии с требуемыми правилами доступа. Однако механизм пользовательских фильтров коммутаторов имеет несколько недостатков:

- Приходится задавать отдельные условия для каждого узла сети, используя при этом громоздкие MAC-адреса. Гораздо проще было бы группировать узлы и описывать условия взаимодействия сразу для групп.
- Невозможно блокировать широковещательный трафик. Широковещательный трафик может быть причиной недоступности сети, если какой-то ее узел умышленно или неумышленно с большой интенсивностью генерирует широковещательные кадры.

Техника виртуальных локальных сетей решает задачу ограничения взаимодействия узлов сети другим способом.

Основное назначение технологии VLAN состоит в облегчении процесса создания изолированных сетей, которые затем обычно связываются между собой с помощью маршрутизаторов. Такое

построение сети создает мощные барьеры на пути нежелательного трафика из одной сети в другую. Сегодня считается очевидным, что любая крупная сеть должна включать маршрутизаторы, иначе потоки ошибочных кадров, например широковещательных, будут периодически «затапливать» всю сеть через прозрачные для них коммутаторы, приводя ее в неработоспособное состояние.

Достоинством технологии виртуальных сетей является то, что она позволяет создавать полностью изолированные сегменты сети, путем логического конфигурирования коммутаторов, не прибегая к изменению физической структуры.

До появления технологии VLAN для создания отдельной сети использовались либо физически изолированные сегменты коаксиального кабеля, либо не связанные между собой сегменты, построенные на повторителях и мостах. Затем эти сети связывались маршрутизаторами в единую составную сеть (рис. 38).

Изменение состава сегментов (переход пользователя в другую сеть, дробление крупных сегментов) при таком подходе подразумевает физическую перекоммутацию разъемов на передних панелях повторителей или на кроссовых панелях, что не очень удобно в больших сетях - много физической работы, к тому же высока вероятность ошибки.

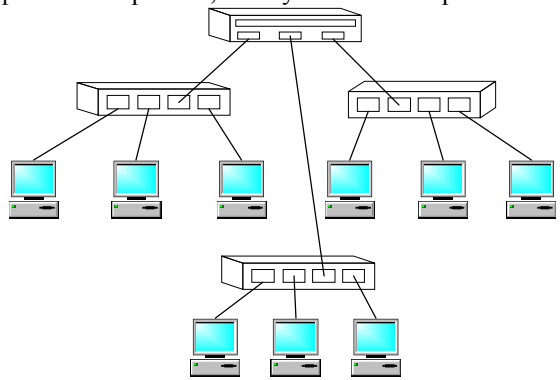


**Рис. 38** Составная сеть, состоящая из сетей, построенных на основе повторителей

Для связывания виртуальных сетей в общую сеть требуется привлечение средств сетевого уровня. Он может быть реализован в отдельном маршрутизаторе или в составе программного обеспечения коммутатора, который тогда становится комбинированным устройством - коммутатором 3-го уровня.

Технология виртуальных сетей долгое время не

стандартизировалась, хотя и была реализована в очень широком спектре моделей коммутаторов разных производителей. Положение изменилось после принятия в 1998 году стандарта IEEE 802.1Q, который определяет базовые правила построения виртуальных локальных сетей, не зависящие от протокола канального уровня, поддерживаемого коммутатором.

## 17.2. Создание виртуальных сетей на базе одного коммутатора

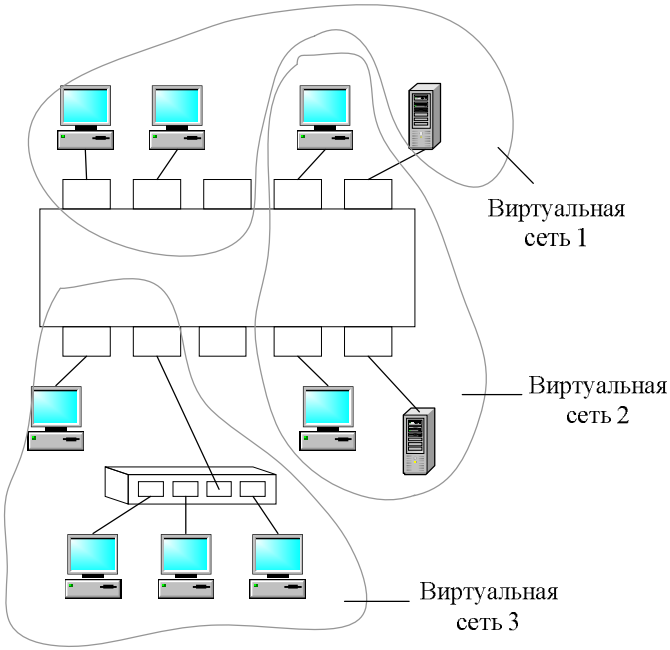



**Рис. 39** Виртуальные сети, построенные на одном коммутаторе

При создании виртуальных сетей на основе одного коммутатора обычно используется механизм группирования портов коммутатора (рис. 39). При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего, например, виртуальной сети 1, никогда не будет передан порту, который не принадлежит этой виртуальной сети. Порт можно приписать нескольким виртуальным сетям, хотя на практике так делают редко — пропадает эффект полной изоляции сетей.

Создание виртуальных сетей путем группирования портов не

требует от администратора большого объема ручной работы — достаточно каждый порт приписать к одной из нескольких заранее поименованных виртуальных сетей. Обычно такая операция выполняется с помощью специальной программы, прилагаемой к коммутатору.

Второй способ образования виртуальных сетей основан на группировании MAC-адресов. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует от администратора большого объема ручной работы. Однако при построении виртуальных сетей на основе нескольких коммутаторов он оказывается более гибким, чем группирование портов.

## 17.3. Создание виртуальных сетей на базе нескольких коммутаторов

Рис. 40 иллюстрирует проблему, возникающую при создании виртуальных сетей на основе нескольких коммутаторов, поддерживающих технику группирования портов:

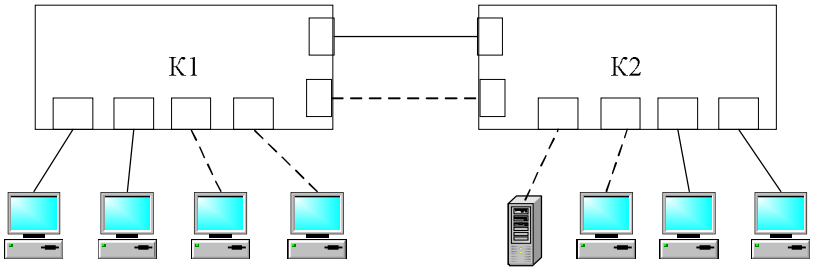



**Рис. 40** Построение виртуальных сетей на нескольких коммутаторах с группированием портов

Если узлы какой-либо виртуальной сети подключены к разным коммутаторам, то для подключения каждой такой сети на коммутаторах должна быть выделена специальная пара портов. Таким образом, коммутаторы с группированием портов требуют для своего соединения столько портов, сколько виртуальных сетей они поддерживают. Порты и кабели используются в этом случае очень неэкономно. Кроме того, при соединении виртуальных сетей через маршрутизатор для каждой виртуальной сети выделяется отдельный кабель и отдельный порт маршрутизатора, что также приводит к большим накладным расходам.

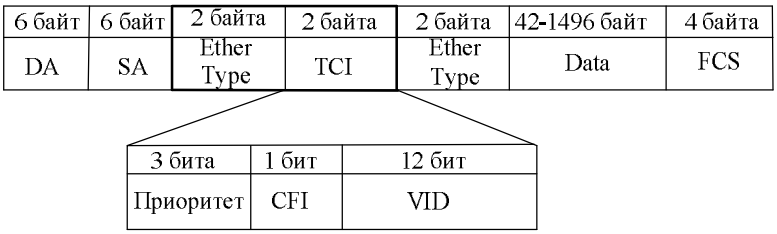
Группирование MAC-адресов в виртуальную сеть на каждом коммутаторе избавляет от необходимости связывать их по нескольким портам, поскольку в этом случае MAC-адрес становится меткой виртуальной сети. Однако этот способ требует выполнения большого количества ручных операций по маркировке MAC-адресов на каждом коммутаторе сети.

Описанные два подхода основаны только на добавлении дополнительной информации к адресным таблицам коммутатора и в них отсутствует возможность встраивания в передаваемый кадр информации о принадлежности кадра виртуальной сети. В остальных подходах используются имеющиеся или дополнительные поля кадра для сохранения информации о принадлежности кадра той или иной виртуальной локальной сети при его перемещениях между коммутаторами сети. При этом нет необходимости запоминать в каждом коммутаторе принадлежность всех MAC-адресов составной сети соответствующим виртуальным сетям.

Дополнительное поле с пометкой о номере виртуальной сети используется только тогда, когда кадр передается от коммутатора к коммутатору, а при передаче кадра конечному узлу оно обычно удаляется. При этом модифицируется протокол взаимодействия «коммутатор-коммутатор», а программное и аппаратное обеспечение конечных узлов остается неизменным.

Ethernet вносит дополнительный заголовок, который называется тегом виртуальной локальной сети.

Тег VLAN




Тег VLAN не является обязательным для кадров Ethernet. Кадр, у которого имеется такой заголовок, называют помеченным (tagged frame). Коммутаторы могут одновременно работать как с помеченными, так и с непомеченными кадрами. Из-за добавления тега VLAN максимальная длина поля данных уменьшилась на 4 байта.

Для того чтобы оборудование локальных сетей могло отличать и понимать помеченные кадры, для них введено специальное значение поля EtherType, равное 0x8100. Это значение говорит о том, что за ним следует

поле TCI, а не стандартное поле данных. Обратите внимание, что в помеченном кадре за полями тега VLAN следует другое поле EtherType, указывающее тип протокола, данные которого переносятся полем данных кадра.

В поле TCI находится 12-битное поле номера (идентификатора) VLAN, называемого VID. Разрядность поля VID позволяет коммутаторам создавать до 4096 виртуальных сетей.

Пользуясь значением VID в помеченных кадрах, коммутаторы сети выполняют групповую фильтрацию трафика, разбивая сеть на виртуальные сегменты, то есть на VLAN. Для поддержки этого режима каждый порт коммутатора приписывается к одной или нескольким виртуальным локальным сетям, то есть выполняется группировка портов.

Для упрощения конфигурирования сети в стандарте 802.1Q появляются понятия линии доступа и транка.

Линия доступа связывает порт коммутатора (называемый в этом случае портом доступа) с компьютером, принадлежащим некоторой виртуальной локальной сети.

Транк – это линия связи, которая соединяет между собой порты двух коммутаторов, в общем случае через транк передается трафик нескольких виртуальных сетей.

Для того чтобы образовать в исходной сети виртуальную локальную сеть, нужно в первую очередь выбрать для нее значение идентификатора VID, отличное от 1, а затем, используя команды конфигурирования коммутатора, приписать к этой сети те порты, к которым присоединены включаемые в нее компьютеры. Порт доступа может быть приписан только к одной виртуальной локальной сети.

Порты доступа получают от конечных узлов сети непомеченные кадры и помечают их тегом VLAN, содержащим то значение VID, которое назначено этому порту. При передаче же помеченных кадров конечному узлу порт доступа удаляет тег виртуальной локальной сети.

Для более наглядного описания вернемся к рассмотренному ранее примеру сети. На рис. 41 показано, как решается задача избирательного доступа к серверам на основе техники VLAN.

Будем считать, что поставлена задача обеспечить доступ компьютеров C1 и C3 к серверам S1 и S3, в то время как компьютеры C2 и C4 должны иметь доступ только к серверам S2 и S4.

Чтобы решить эту задачу, нужно организовать в сети две виртуальные локальные сети, VLAN2 и VLAN3 (напомним, что сеть VLAN1 уже существует по умолчанию — это наша исходная сеть), приписать один набор компьютеров и серверов к VLAN2, а другой – к VLAN3.

Для приписывания конечных узлов к определенной виртуальной локальной сети соответствующие порты объявляются портами доступа этой сети путем назначения им соответствующего идентификатора VID. Например, порт 1 коммутатора SW1 должен быть объявлен портом доступа VLAN2 путем назначения ему идентификатора VID2, то же самое должно быть проделано с портом 5 коммутатора SW1, портом 1 коммутатора SW2, портом 1 коммутатора SW3. Порты доступа сети VLAN3 должны получить идентификатор VID3.

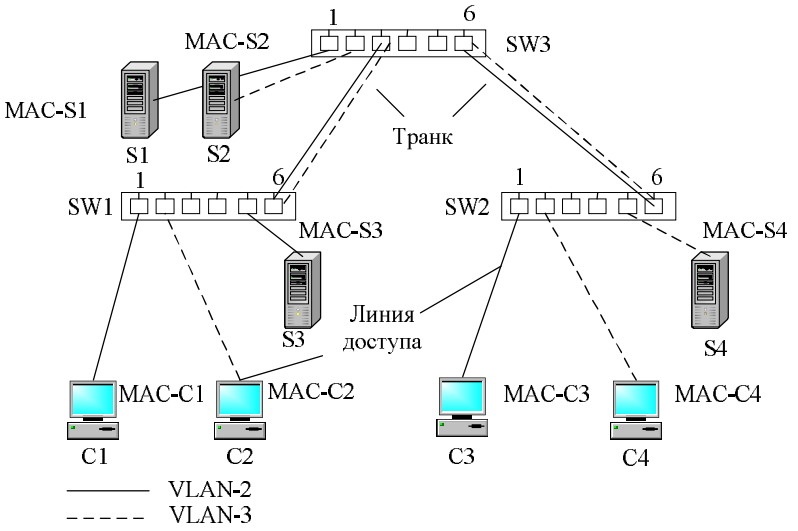



**Рис. 41** Разбиение сети на две виртуальные локальные сети

В нашей сети нужно также организовать транки — те линии связи, которые соединяют между собой порты коммутаторов. Порты, подключенные к транкам, не добавляют и не удаляют теги, они просто передают кадры в неизменном виде. В нашем примере такими портами должны быть порты 6 коммутаторов SW1 и SW2, а также порты 3 и 6 коммутатора SW3. Порты в нашем примере должны поддерживать сети VLAN2 и VLAN3 (и VLAN1, если в сети есть узлы, явно не приписанные ни к одной виртуальной локальной сети).

Коммутаторы, поддерживающие технологию VLAN, осуществляют дополнительную фильтрацию трафика. В том случае если таблица продвижения коммутатора говорит о том, что пришедший кадр нужно передать на некоторый порт, перед передачей коммутатор проверяет, соответствует ли значение VTD в теге VLAN кадра той виртуальной

локальной сети, которая приписана к этому порту. В случае соответствия кадр передается, несоответствия — отбрасывается. Непомеченные кадры обрабатываются аналогичным образом, но с использованием условной сети VLAN1. MAC-адреса изучаются коммутаторами сети отдельно.

Техника VLAN оказывается весьма эффективной для разграничения доступа к серверам. Конфигурирование виртуальной локальной сети не требует знания MAC-адресов узлов, кроме того, любое изменение в сети, например подключение компьютера к другому коммутатору, требует конфигурирования лишь порта данного коммутатора, а все остальные коммутаторы сети продолжают работать без внесения изменений, в их конфигурации.

## 18. Стек протоколов TCP/IP

Сегодня стек TCP/IP широко используется как в глобальных, так и в локальных сетях. Этот стек имеет иерархическую структуру, в которой определено 4 уровня (рис. 42):

| | |
|---|---|
| Прикладной уровень | FTP, Telnet, HTTP, SMTP, SNMP, TFTP |
| Транспортный уровень | TCP, UDP |
| Сетевой уровень | IP, ICMP, RIP, OSPF |
| Уровень сетевых интерфейсов | Не регламентируется |

**Рис. 42** Иерархическая структура стека TCP/IP

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет сервисы, предоставляемые системой пользовательским приложениям. Распространены протоколы верхнего уровня, такие как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала Telnet, простой протокол передачи почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (Hypertext Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах.

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- гарантированную доставку обеспечивает протокол управления передачей (Transmission Control Protocol, TCP);

- доставку по возможности, или с максимальными усилиями, обеспечивает протокол пользовательских дейтаграмм (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных, протокол TCP предусматривает установление логического соединения, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Благодаря этому протоколу объекты на хосте-отправителе и хосте-получателе могут поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформированный на одном из компьютеров поток байтов на любой другой компьютер, входящий в составную сеть.

Второй протокол этого уровня, UDP, является простейшим дейтаграммным протоколом, который используется тогда, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

Сетевой уровень, называемый также уровнем Интернета, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением нескольких подсетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов.

Основным протоколом сетевого уровня является межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями от одного маршрутизатора к другому до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней, протокол IP развертывается не только на хостах, но и на всех маршрутизаторах (шлюзах). Протокол IP — это дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями. Такой тип сетевого сервиса называют также «ненадежным».

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации: RIP и OSPF, предназначенные для изучения топологии сети, определения маршрутов и составления таблиц маршрутизации, на основании

которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику сведений об ошибках, возникших при передаче пакета, и некоторые другие протоколы.

У нижнего уровня стека TCP/IP задача существенно проще, чем в модели OSI/ISO. Он отвечает только за организацию взаимодействия с подсетями разных технологий, входящими в составную сеть. TCP/IP рассматривает любую подсеть, входящую в составную сеть, как средство транспортировки пакетов между двумя соседними маршрутизаторами.

Задачу организации интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести к двум задачам:

- упаковка (инкапсуляция) IP-пакета в единицу передаваемых данных промежуточной сети;
- преобразование сетевых адресов в адреса технологии данной промежуточной сети.

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 43):

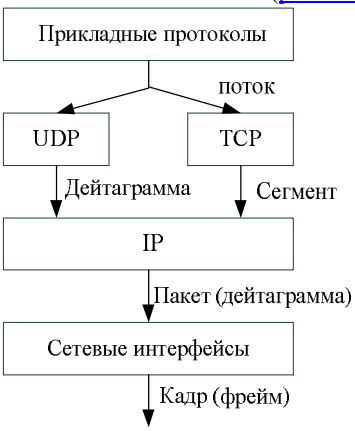



**Рис. 43** Названия протокольных единиц данных в TCP/IP

Потоком данных, информационным потоком, или просто

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

Протокол TCP «нарезает» из потока данных сегменты.

Единицу данных протокола UDP часто называют дейтаграммой, или датаграммой. Дейтаграмма — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных иногда тоже называют дейтаграммой, хотя достаточно часто используется и другой термин — пакет.

В стеке TCP/IP единицы данных любых технологий, в которые упаковываются IP-пакеты для их последующей передачи через сети составной сети, принято называть также кадрами, или фреймами.

## 19. Формат IP-адреса

В заголовке IP-пакета для хранения IP-адресов отправителя и получателя отводятся два поля, каждое имеет фиксированную длину 4 байта (32 бита). IP-адрес состоит из двух логических частей — номера сети и номера узла в сети.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30.

Этот же адрес может быть представлен в двоичном формате:

10000000 00001010 00000010 00011110,

а также в шестнадцатеричном формате:

80.0A.02.1D.

Заметим, что запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла. Вместе с тем при передаче пакета по сети часто возникает необходимость разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор, получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети.

Можно предложить несколько вариантов решения этой проблемы:

- Простейший из них состоит в использовании фиксированной границы. При этом всё 32-битное поле адреса заранее делится на две части не обязательно равные, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, в

другой — номер узла. Поскольку поле, которое отводится для хранения номера узла, имеет фиксированную длину, все сети будут иметь одинаковое максимальное число узлов. Если, например, под номер сети отвести один первый байт, то всё адресное пространство распадется на сравнительно небольшое (28) число сетей огромного размера (234 узлов). Если границу передвинуть дальше вправо, то сетей станет больше, но все равно все они будут одинакового размера. Именно поэтому он не нашел применения.

- Второй подход (RFC 950, RFC 1518) основан на использовании маски, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла. При таком подходе адресное пространство можно использовать для создания множества сетей разного размера.

Маска — это число, применяемое в паре с IP-адресом, причем двоичная запись маски содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностями единиц и нулей в маске соответствует границе между номером сети и номером узла в IP-адресе.

- Наконец, способ, основанный на классах адресов (RFC 791). Этот способ представляет собой компромисс по отношению к двум предыдущим: размеры сетей хотя и не могут быть произвольными, как при использовании масок, но и не должны быть одинаковыми, как при установлении фиксированных границ. Вводится пять классов адресов: A, B, C, D, E. Три из них — A, B и C — предназначены для адресации сетей, а два — D и E — имеют специальное назначение. Для каждого класса сетевых адресов определено собственное положение границы между номером сети и номером узла.

## 19.1. Классы IP-адресов

Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса. Таблица 2 иллюстрирует структуру, IP-адресов разных классов.

К **классу A** относится адрес, в котором старший бит имеет значение 0. В адресах класса A под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Сети, все IP-адреса которых имеют значение первого байта в

диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса A. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей (см. далее). Сетей класса A сравнительно немного, зато количество узлов в них может достигать 224, то есть 16 777 216 узлов.

**Таблица 2** Классы IP-адресов

| Класс | Первые биты | Наименьший номер сети | Наибольший номер сети | Максимальное число узлов в сети |
|---|---|---|---|---|
| A | 0 | 1.0.0.0 (0 – не используется) | 126.0.0.0 (127 – зарезервирован) | 224, поле 3 байт |
| B | 10 | 128.0.0.0 | 191.255.0.0 | 216, поле 2 байт |
| C | 110 | 192.0.0.0 | 223.255.255.0 | 28, поле 1 байт |
| D | 1110 | 224.0.0.0 | 239.255.255.255 | Групповые адреса |
| E | 11110 | 240.0.0.0 | 247.255.255.255 | Зарезервировано |

К **классу B** относятся все адреса, старшие два бита которых имеют значение 10. В адресах класса B под номер сети и под номер узла отводится по 2 байта. Сети, значения первых двух байтов адресов которых находятся в диапазоне от 128.0 (10000000 00000000) до 191.255 (1011111111111111), называются сетями класса B. Ясно, что сетей класса B больше, чем сетей класса A, а размеры их меньше. Максимальное количество узлов в сетях класса B составляет 216 (65 536).

К **классу C** относятся все адреса, старшие три бита которых имеют значение 110. В адресах класса C под номер сети отводится 3 байта, а под номер узла — 1 байт. Сети, старшие три байта которых находятся в диапазоне от 192.0.0 (11000000 00000000 00000000) до 223.255.255 (11011111 11111111 11111111), называются сетями класса C. Сети класса C наиболее распространены, и наименьшее максимальное число узлов в них равно 28(256).

Если адрес начинается с последовательности 1110, то он является адресом **класса D** и обозначает особый групповой адрес (multicast address). В то время как адреса классов A, B и C служат для идентификации отдельных сетевых интерфейсов, то есть являются индивидуальными адресами (unicast address), групповой адрес идентифицирует группу сетевых интерфейсов, которые в общем случае могут принадлежать разным сетям. Интерфейс, входящий в группу, получает наряду с обычным индивидуальным IP-адресом еще один групповой адрес. Если при отправке пакета в качестве адреса назначения указан адрес класса D, то такой пакет должен быть

доставлен всем узлам, которые входят в группу.

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

Чтобы получить из IP-адреса номер сети и номер узла, требуется не только разделить адрес на две соответствующие части, но и дополнить каждую из них нулями до 4 полных байт. Возьмем, например, адрес класса B 129.64.134.5. Первые два байта идентифицируют сеть, а последующие два — узел. Таким образом, номером сети является адрес 129.64.0.0, а номером узла — адрес 0.0.134.5.

## 19.2. Особые IP-адреса

В стеке TCP/IP существуют ограничения при назначении IP-адресов, а именно номера сетей и номера узлов не могут состоять из одних двоичных нулей или единиц. Отсюда следует, что максимальное количество узлов, приведенное в табл. 2 для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса C под номер узла отводится 8 бит, которые позволяют задать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 запрещены для адресации сетевых интерфейсов. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Итак, некоторые IP-адреса интерпретируются особым образом:

**1.** Если IP-адрес состоит только из двоичных нулей, то он называется неопределенным адресом и обозначает адрес того узла, который сгенерировал этот пакет. Адрес такого вида в особых случаях помещается в заголовок IP-пакета в поле адреса отправителя.

**2.** Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет. Такой адрес также может быть использован только в качестве адреса отправителя.

**3.** Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется ограниченным широковещательным (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной сети не при каких условиях.

**4.** Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем узлам сети 192.190.21.0. Такой тип адреса называется широковещательным (broadcast).

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является внутренним адресом стека протоколов компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети.

В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со значения 127. Когда программа посылает данные по IP-адресу 127.x.x.x, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется адресом обратной петли (loopback).

## 19.3. Использование масок при IP-адресации

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации.

Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде IP-адрес 129.64.134.5 — это:

10000001.01000000.10000110.00000101,

а маска 255.255.128.0 в двоичном виде выглядит так:

11111111.11111111.10000000.00000000.

Если игнорировать маску и интерпретировать адрес 129.64.134.5 на основе классов, то номером сети является 129.64.0.0, а номером узла — 0.0.134.5 (поскольку адрес относится к классу B).

Если же использовать маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес 129.64.134.5, делят его на две части, номер сети:

10000001.01000000.1

и номер узла:

0000110.00000101.

В десятичной форме записи номера сети и узла, дополненные нулями до 32 бит, выглядят соответственно как 129.64.128.0 и 0.0.6.5.

Наложение маски можно интерпретировать как выполнение логической операции И (AND). Так, в предыдущем примере номер сети из адреса 129.64.134.5 является результатом выполнения логической операции AND с маской 255.255.128.0:

10000001 01000000 10000110 00000101

AND

11111111.11111111.10000000.00000000.

Для стандартных классов сетей маски имеют следующие значения:

- класс А - 11111111.00000000.00000000. 00000000 (255.0.0.0);
- класс В - 11111111.11111111. 00000000. 00000000 (255.255.0.0);
- класс С - 11111111. 11111111.11111111.00000000 (255.255.255.0).

Механизм масок широко распространен в маршрутизации IP, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбивать одну, выделенную ему поставщиком услуг сеть определенного класса на несколько других, не требуя от него дополнительных номеров сетей — эта операция называется разделением на подсети (subnetting).

## 20. Порядок назначения IP-адресов

По определению схема IP-адресации должна обеспечивать уникальность нумерации сетей, а также уникальность нумерации узлов в пределах каждой из сетей.

### 20.1. Назначение адресов автономной сети

В небольшой автономной IP-сети условие уникальности номеров сетей и узлов может быть выполнено силами сетевого администратора.

В этом случае в распоряжении администратора имеется все адресное пространство, так как совпадение IP-адресов в не связанных между собой сетях не вызовет никаких отрицательных последствий.

Однако при таком подходе исключена возможность в будущем подсоединить данную сеть к Интернету. Действительно, произвольно выбранные адреса данной сети могут совпасть с централизованно назначенными адресами Интернета. Для того чтобы избежать коллизий, связанных с такого рода совпадениями, в стандартах Интернета определено несколько диапазонов так называемых частных адресов, рекомендуемых для автономного использования:

- в классе A сеть 10.0.0.0;
- в классе B диапазон из 16 номеров сетей (172.16.0.0-172.31.0.0);
- в классе C диапазон из 255 сетей (192.168.0.0-192.168.255.0).

Эти адреса, исключенные из множества централизованно распределяемых, составляют огромное адресное пространство, достаточное для нумерации узлов автономных сетей практически любых размеров. Частные адреса, как и при произвольном выборе адресов, в разных автономных сетях могут совпадать. В то же время использование частных адресов для адресации автономных сетей делает возможным корректное подключение их к Интернету. Применяемые при этом специальные технологии подключения исключают коллизии адресов.

## 20.2. Централизованное распределение адресов

В больших сетях, подобных Интернету, уникальность сетевых адресов гарантируется централизованной, иерархически организованной системой их распределения. Номер сети может быть назначен только по рекомендации специального подразделения Интернета.

Главным органом регистрации глобальных адресов в Интернете с 1998 года является неправительственная некоммерческая организация ICANN (Internet Corporation for Assigned Names and Numbers). Эта организация координирует работу региональных отделов, деятельность которых охватывает большие географические площади: ARIN — Америка, RIPE (Европа), APNIC (Азия и Тихоокеанский регион). Региональные отделы выделяют блоки адресов сетей крупным поставщикам услуг, а те, в свою очередь, распределяют их между своими клиентами, среди которых могут быть и более мелкие поставщики.

Проблемой централизованного распределения адресов является их дефицит. Уже сравнительно давно очень трудно получить адрес класса B и практически невозможно стать обладателем адреса класса A. При этом надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся адресное пространство используется нерационально. Очень часто владельцы сетей класса C расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве линии связи используют два маршрутизатора, соединенных по двухточечной схеме (рис. 44). Для вырожденной сети,

образованной линией связи, связывающей порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети всего два узла.

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию протокола IP — протокол IPv6, в котором резко расширяется адресное пространство.

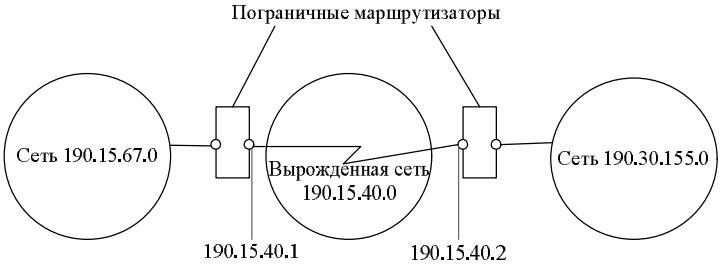



**Рис. 44** Нерациональное использование пространства IP-адресов

## 21. Адресация и технология CIDR

Деление IP-адреса на номера сети и узла в технологии CIDR происходит на основе маски переменной длины, назначаемой поставщиком услуг. Непременным условием применимости CIDR является наличие у организации, распоряжающейся адресами, непрерывных диапазонов адресов. Такие адреса имеют одинаковый префикс, то есть одинаковую цифровую последовательность в нескольких старших разрядах. Пусть в распоряжении некоторого поставщика услуг имеется непрерывное пространство IP-адресов в количестве 2n (рис. 45). Отсюда следует, что префикс имеет длину (32 - n) разрядов. Остальные n разрядов играют роль счетчика последовательных номеров.

Когда потребитель обращается к поставщику услуг с просьбой о выделении ему некоторого числа адресов, то в имеющемся пуле адресов «вырезается» непрерывная область S1,S2 или S3, в зависимости от требуемого количества адресов. При этом должны быть выполнены следующие условия:

- количество адресов в выделяемой области должно быть равно степени двойки;
- начальная граница выделяемого пула адресов должна быть кратна требуемому количеству узлов.

Очевидно, что префикс каждой из показанных на рисунке

областей имеет собственную длину — чем меньше количество адресов в данной области, тем длиннее ее префикс.

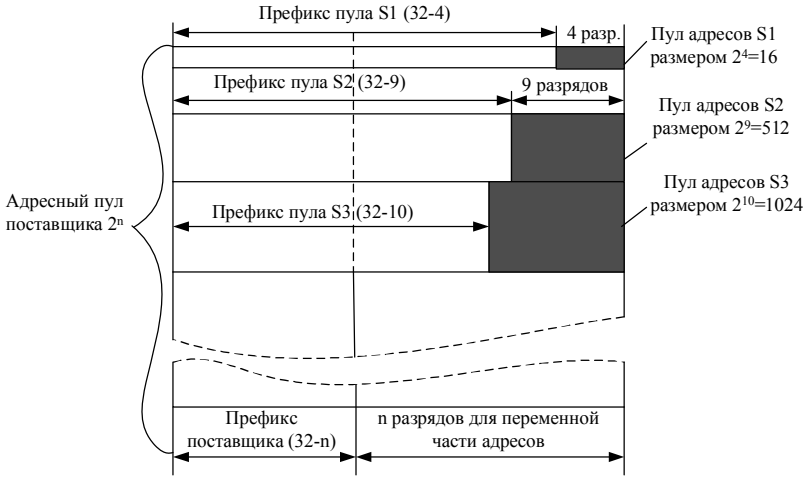



**Рис. 45** Распределение адресов на основе технологии CIDR

**ПРИМЕР**

Пусть поставщик услуг Интернета располагает пулом адресов в диапазоне 193.20.0.0-193.23.255.255 (1100 0001.0001 0100.0000 0000.0000 0000-1100 0001.0001 0111.1111 1111.1111 1111), то есть количество адресов равно 218. Соответственно префикс поставщика услуг имеет длину 14 разрядов — 1100 0001.0001 01, или в другом виде — 193.20/14.

Если абоненту этого поставщика услуг требуется совсем немного адресов, например 13, то поставщик мог бы предложить ему различные варианты: сеть 193.20.30.0/28, сеть 193.20.30.16/28 или сеть 193.21.204.48/28. Во всех случаях в распоряжении абонента для нумерации узлов имеются 4 младших бита. Таким образом, наименьшее число, удовлетворяющее потребностям абонента (13), которое можно представить степенью двойки (24), является 16. Префикс для каждого из выделяемых пулов во всех этих случаях играет роль номера сети, он имеет длину 32-4=28 разрядов.

Рассмотрим другой вариант, когда к поставщику услуг обратился крупный заказчик, сам, возможно, собирающийся оказывать услуги по доступу в Интернет. Ему требуется блок адресов в 4000 узлов. На нумерацию такого количества узлов пойдет 12 двоичных разрядов, следовательно, размер выделенного пула адресов оказывается

несколько больше требуемого - 4096. Граница, с которой должен начинаться выделяемый участок, должна быть кратна размеру участка, то есть это могут быть любые адреса из следующих: 193.20.0.0,193.20.16.0,193.20.32.0, 193.20.48.0 или др. Пусть поставщик услуг предложил потребителю диапазон адресов 193.20.10.0 - 193.20.31.253. Для этого диапазона агрегированный номер сети (префикс) имеет длину 20 двоичных разрядов и равен 193.20.16.0/20.

Благодаря CIDR поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в соответствии с действительными требованиями каждого клиента.

# 22. Отображение IP-адресов на локальные адреса

## 22.1 Протокол разрешения адресов

Никакой функциональной зависимости между локальным адресом и его IP-адресом не существует, следовательно, единственный способ установления соответствия — ведение таблиц. В результате конфигурирования сети каждый интерфейс «знает» свой IP-адрес и локальный адрес, что можно рассматривать как таблицу, состоящую из одной строки. Проблема состоит в том, как организовать обмен имеющейся информацией между узлами сети.

Для определения локального адреса по IP-адресу используется протокол разрешения адресов (Address Resolution Protocol, ARP). Протокол разрешения адресов реализуется различным образом в зависимости от того, работает ли в данной сети протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещания или же какой-либо из протоколов глобальной сети (Frame Relay, ATM), которые, как правило, не поддерживают широковещательный доступ.

Рассмотрим работу протокола ARP в локальных сетях с широковещанием.

На рис. 46 показан фрагмент IP-сети, включающий две сети — Ethernet1 (из трех конечных узлов A, B и C) и Ethernet2 (из двух конечных узлов D и E). Сети подключены соответственно к интерфейсам 1 и 2 маршрутизатора. Каждый сетевой интерфейс имеет IP-адрес и MAC-адрес. Пусть в какой-то момент IP-модуль узла C направляет пакет узлу D. Протокол IP узла C определил IP-адрес интерфейса следующего маршрутизатора — это IP1. Теперь, прежде чем упаковать пакет в кадр Ethernet, и направить его маршрутизатору, необходимо определить соответствующий MAC-адрес. Для решения этой задачи протокол IP обращается к протоколу ARP. Протокол ARP

поддерживает на каждом интерфейсе сетевого адаптера или маршрутизатора отдельную ARP-таблицу, в которой в ходе функционирования сети накапливается информация о соответствии между IP-адресами и MAC-адресами других интерфейсов данной сети. Первоначально, при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

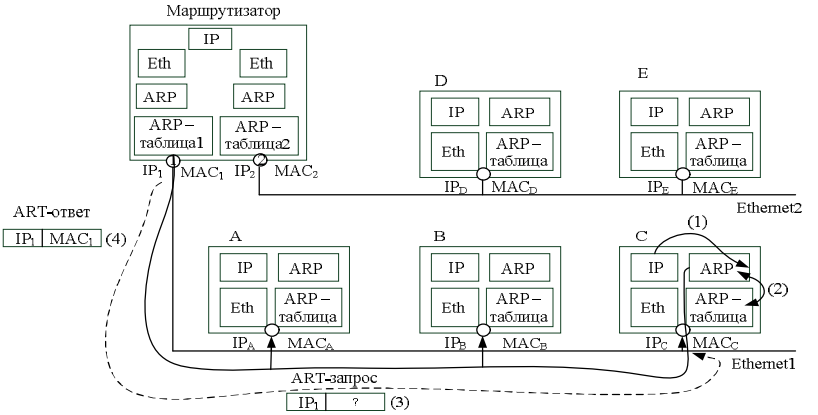



**Рис. 46** Схема работы протокола ARP

**1.** На первом шаге происходит передача от протокола IP протоколу ARP примерно такого сообщения: «Какой MAC-адрес имеет интерфейс с адресом IP1?».

**2.** Работа протокола ARP начинается с просмотра собственной ARP-таблицы. Предположим, что среди содержащихся в ней записей отсутствует запрашиваемый IP-адрес.

**3.** В этом случае исходящий IP-пакет, для которого оказалось невозможным определить локальный адрес из ARP-таблицы, запоминается в буфере, а протокол ARP формирует ARP-запрос, вкладывает его в кадр протокола Ethernet и широковещательно рассылает.

**4.** Все интерфейсы сети Ethernet1 получают ARP-запрос и направляют его «своему» протоколу ARP. ARP сравнивает указанный в запросе адрес IP1 с IP-адресом интерфейса, на который поступил этот запрос. Протокол ARP, который констатировал совпадение (в данном случае это ARP маршрутизатора 1), формирует ARP-ответ.

В ARP-ответе маршрутизатор указывает локальный адрес MAC1 своего интерфейса и отправляет его запрашивающему узлу (в данном примере узлу C), используя его локальный адрес. Широковещательный ответ в этом случае не требуется, так как формат ARP-запроса

предусматривает поля локального и сетевого адресов отправителя. Заметим, что зона распространения ARP-запросов ограничивается сетью Ethernet1, так как на пути широковещательных кадров барьером стоит маршрутизатор.

На рис. 47 показан кадр Ethernet с вложенным в него ARP-сообщением. ARP-запросы и ARP-ответы имеют один и тот же формат. В табл. 3 в качестве примера приведены значения полей реального ARP-запроса, переданного по сети Ethernet.

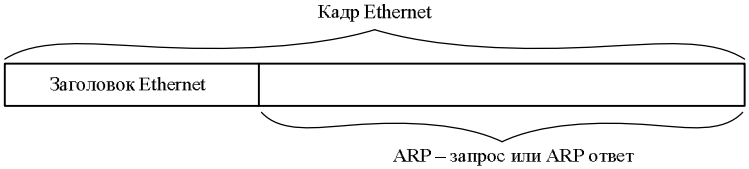



**Рис. 47** Инкапсуляция ARP-сообщений в кадр Ethernet

В поле типа сети для сетей Ethernet указывается значение 1. Поле типа протокола позволяет использовать протокол ARP не только с протоколом IP, но и с другими сетевыми протоколами. Для IP значение этого поля равно 0x0800. Длина локального адреса для протокола Ethernet равна 6 байт, а длина IP-адреса — 4 байта. В поле операции для ARP-запросов указывается значение 1, для ARP-ответов — значение 2.

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

**Таблица 3** Пример ARP-запроса

| Поле | Значение |
|---|---|
| Тип сети | 1(0x1) |
| Тип протокола | 2048 (0x800) |
| Длина локального адреса | 6 (0x6) |
| Длина сетевого адреса | 4 (0x4) |
| Операция | 1 (0x1) |
| Локальный адрес отправителя | 00804SEB7E60 |
| Сетевой адрес отправителя | 194.85.135.75 |
| Локальный (искомый) адрес получателя | 000000000000 |
| Сетевой адрес получателя | 194.85.135.65 |

Ответ присылает узел, опознавший свой IP-адрес. Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет. Протокол IP уничтожает IP-пакеты, направляемые по этому адресу. В табл. 4 показаны значения полей ARP-ответа, который мог бы поступить на приведенный в табл. 3 ARP-запрос.

**Таблица 4** Пример ARP-ответа

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

В результате обмена ARP-сообщениями модуль IP, пославший запрос с интерфейса, имеющего адрес 194.85.135.75, определил, что IP-адресу 194.85.135.65 соответствует MAC-адрес 00E0F77F1920. Этот адрес затем помещается в заголовок кадра Ethernet, ожидавшего отправления IP-пакета.

Чтобы уменьшить число ARP-обращений в сети, найденное соответствие между IP-адресом и MAC-адресом сохраняется в ARP-таблице соответствующего интерфейса, в данном случае — это запись:

194.85.135.65 - 00EOF77F1920.

Данная запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как модуль ARP проанализирует ARP-ответ. Теперь, если вдруг вновь возникнет необходимость послать пакет по адресу 194.85.135.65, то протокол IP прежде чем посылать широковещательный запрос, проверит, нет ли уже такого адреса в ARP-таблице. ARP-таблица пополняется не только за счет поступающих на данный интерфейс ARP-ответов, но и в результате извлечения полезной информации из широковещательных ARP-запросов. Действительно, в каждом запросе, как это видно из табл. 3 и 4, содержатся IP-адрес и MAC-адрес отправителя. Все интерфейсы, получившие этот запрос, могут поместить информацию о соответствии локального и сетевого адресов отправителя в собственную ARP-таблицу. В частности, все узлы, получившие ARP-запрос (см. табл. 3), могут пополнить свою ARP-таблицу записью: 194.85.135.75 - 008048EB7E60.

Таким образом, вид ARP-таблицы, в которую в ходе работы сети были добавлены две упомянутые нами записи, иллюстрирует табл. 5.

**Таблица 5** Пример ARP-таблицы

| IP-адрес | MAC-адрес | Тип записи |
|---|---|---|
| 194.85.135.65 | 00E0F77F190 | Динамический |
| 194.85.135.75 | 008048EB7E60 | Динамический |
| 104.85.60.21 | 008048EB7567 | Статический |

В ARP-таблицах существует два типа записей: динамические и статические. Статические записи создаются вручную с помощью утилиты arp и не имеют срока устаревания, точнее, они существуют до тех пор, пока компьютер или маршрутизатор остается включенным. Динамические записи должны периодически обновляться. Если запись не обновлялась в течение определенного времени (порядка нескольких минут), то она исключается из таблицы. Таким образом, в ARP-таблице содержатся записи не обо всех узлах сети, а только о тех, которые активно участвуют в сетевых операциях. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют ARP-кэшем.

В некоторых случаях возникает обратная задача — нахождение IP-адреса по известному локальному адресу. Тогда в действие вступает реверсивный протокол разрешения адресов (Reverse Address Resolution Protocol, RARP). Этот протокол используется, например, при старте бездисковых станций, не знающих в начальный момент времени своего IP-адреса, но знающих MAC-адрес своего сетевого адаптера.

## 23. Формат IP-пакета

IP-пакет состоит из полей заголовка и данных. Далее перечислены поля заголовка:


**Рис. 48** Структура заголовка IP-пакета

Поле номера версии занимает 4 бита и идентифицирует версию протокола IP.

Значение длины заголовка IP-пакета также занимает 4 бита и измеряется в 32-битных словах. Обычно заголовок имеет длину в 20 байт (пять 32-битных слов), но при добавлении некоторой служебной информации это значение может быть увеличено за счет дополнительных байтов в поле параметров. Наибольшая длина заголовка составляет 60 байт.

Поле типа сервиса (Type of Service, ToS) — байт дифференцированного обслуживания, или DS-байт. Этим двум названиям соответствуют два варианта интерпретации этого поля. В обоих случаях данное поле служит одной цели - хранению признаков, которые отражают требования к качеству обслуживания пакета. В прежнем варианте первые три бита содержат значение приоритета пакета: от самого низкого — 0 до самого высокого 7. Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Следующие три бита поля ToS определяют критерий выбора маршрута. Если бит D (Delay — задержка) установлен в 1, то маршрут должен выбираться для минимизации задержки доставки данного пакета, установленный бит T (Throughput — пропускная способность) — для максимизации пропускной способности, а бит R (Reliability — надежность) — для максимизации надежности доставки. Оставшиеся два бита имеют нулевое значение.

Поле общей длины занимает 2 байта и характеризует общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля определяющего эту величину, и составляет 65 535 байт, однако в большинстве компьютеров и сетей столь большие пакеты не используются.

Идентификатор пакета занимает 2 байта и используется для распознавания пакетов образовавшихся путем деления на части (фрагментации) исходного пакета. Все части (фрагменты) одного пакета должны иметь одинаковое значение этого поля.

Флаги занимают 3 бита и содержат признаки, связанные с фрагментацией. Установленный в 1 бит DF (Do not Fragment - не фрагментировать) запрещает маршрутизатору фрагментировать данный пакет, а установленный в 1 бит MF (More Fragments — больше фрагментов) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле смещения фрагмента занимает 13 бит и задает смещение в

байтах поля данных этого фрагмента относительно начала поля данных исходного (не фрагментированного) пакета, используется при сборке/разборке фрагментов пакетов. Смещение должно быть кратно 8 байтам.

Поле времени жизни (Time To Live, TTL) занимает один байт и используется для задания предельного срока, в течение которого пакет может перемещаться по сети. Время жизни пакета измеряется в секундах и задается источником. По истечении каждой секунды пребывания на каждом из маршрутизаторов, через которые проходит пакет во время своего «путешествия» по сети, из его текущего времени жизни вычитается единица; единица вычитается и в том случае, если время пребывания было меньше секунды. Если значение поля времени жизни становится нулевым до того, как пакет достигает получателя, пакет уничтожается.

Поле протокола верхнего уровня занимает один байт и содержит идентификатор, указывающий, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета. Например, 6 означает, что в пакете находится сообщение протокола TCP, 17 — протокола UDP, 1 — протокола ICMP.

Контрольная сумма заголовка занимает 2 байта (16 бит) и рассчитывается только для заголовка. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, поле времени жизни), контрольная сумма проверяется и повторно рассчитывается на каждом маршрутизаторе и конечном узле как дополнение к сумме всех 16-битных слов заголовка. При вычислении контрольной суммы значение самого поля контрольной суммы устанавливается в нуль. Если контрольная сумма неверна, то пакет отбрасывается, как только обнаруживается ошибка. Поля IP-адресов источника и приемника имеют одинаковую длину — 32 бита.

Поле параметров является необязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей одного из восьми предопределенных типов.

Далее приведена распечатка значений полей заголовка одного из реальных IP-пакетов, захваченных в сети Ethernet средствами анализатора протоколов сетевого монитора (Network Monitor, NM) компании Microsoft. В данной распечатке NM в скобках указаны шестнадцатеричные значения полей, кроме того, программа иногда представляет числовые коды полей в виде, более удобном для чтения. Например, дружественный программный интерфейс NM интерпретирует код 6 в поле протокола, помещая туда название соответствующего протокола — TCP (см. строку, выделенную

полужирным шрифтом).

**Распечатка значений полей заголовка одного из реальных IP-пакетов**

IP: Version = 4 (0x4)
IP: Header Length = 20 (0x14)
IP: Service Type = 0 (0x0)
IP: Precedence = Routine
IP: ...0….- Normal Delay
IP: ....0... = Normal Throughput
IP: …..0.. = Normal Reliability
IP: Total Length = 54 (0x36)
IP: Identification = 31746 (0x7C02)
IIP: Flags Summary = 2 (0x2)
IP: …….0 = Last fragment in datagram
IP: ……1. = Cannot fragment datagram
IP: Fragment Offset = 0 (0x0) bytes
IP: Time to Live = 128 (0x80)
**IP: Protocol = TCP - Transmission Control**
IP: Checksum = 0xEB86
IP: Source Address = 194.85.135.75
Ip: Destination Address = 194.85.135.66
IP: Data: Number of data bytes remaining = 34 (0x0022)

## 24. Схема IP-маршрутизации

Рассмотрим механизм IP-маршрутизации на примере составной сети, представленной на рис. 49. В этой сети 20 маршрутизаторов (изображенных в виде пронумерованных квадратных блоков) объединяют 18 сетей в общую сеть. N1, N2,..., N18 — это номера сетей. На каждом маршрутизаторе и конечных узлах A и B функционируют протоколы IP.

К нескольким интерфейсам (портам) маршрутизаторов присоединяются сети. Каждый интерфейс маршрутизатора можно рассматривать как отдельный узел сети: он имеет сетевой адрес и локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор под номером 1 имеет три интерфейса, к которым подключены сети N1, N2, N3. На рисунке сетевые адреса этих портов обозначены IP11, IP12 и IP13. Интерфейс IP11 является узлом сети N1, и, следовательно, в поле номера сети порта IP11 содержится номер N1. Аналогично интерфейс IP21 — это узел в сети N2, а порт

IP13 — узел в сети N3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет выделенного адреса, ни сетевого, ни локального.

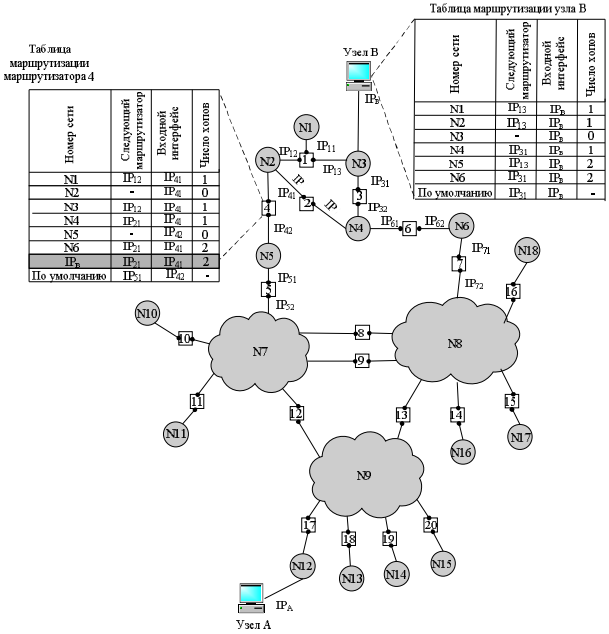



**Рис. 49** Принципы маршрутизации в составной сети

В сложных составных сетях почти всегда существуют несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Так, пакет, отправленный из узла А в узел В, может пройти через маршрутизаторы 17,12,5,4 и 1 или маршрутизаторы 17,13,7,6 и 3. Нетрудно найти еще несколько маршрутов между узлами А и В.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании критерия выбора маршрута. В качестве критерия часто выступает задержка прохождения маршрута отдельным пакетом, средняя пропускная способность маршрута для последовательности пакетов или наиболее простой критерий, учитывающий только количество пройденных на маршруте промежуточных маршрутизаторов (ретрансляционных участков, или хопов). Полученная в результате анализа информация о маршрутах

дальнейшего следования пакетов помещается в таблицу маршрутизации.

## 24.1. Упрощенная таблица маршрутизации

**Таблица 6** Таблица маршрутизации маршрутизатора 4

| Адрес назначения | Сетевой адрес следующего | Сетевой адрес | Расстояние до сети назначения |
|---|---|---|---|
| N1 | $IP_{12}$(R1) | $IP_{41}$ | 1 |
| N2 | — | $IP_{41}$ | 0(подсоединена) |
| N3 | $IP_{12}$(R1) | $IP_{41}$ | 1 |
| N4 | $IP_{21}$(R2) | $IP_{41}$ | 1 |
| N5 | — | $IP_{42}$ | 0(подсоединена) |
| N6 | $IP_{21}$(R2) | $IP_{21}$ | 2 |
| $IP_B$ | $IP_{21}$(R2) | $IP_{41}$ | 2 |
| Маршрут по | $IP_{51}$(R5) | $IP_{42}$ | — |

Первый столбец таблицы содержит адреса назначения пакетов. В каждой строке таблицы, следом за адресом назначения, указывается сетевой адрес следующего маршрутизатора (точнее, сетевой адрес интерфейса следующего маршрутизатора) на который надо направить пакет, чтобы тот передвигался по направлению к заданному адресу по рациональному маршруту.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов ($IP_{41}$ или $IP_{42}$) он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации, содержащий сетевые адреса выходных интерфейсов.

Когда пакет поступает на маршрутизатор, модуль IP извлекает из его заголовка адрес сети назначения и последовательно сравнивает его с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети показывает ближайший маршрутизатор, на который следует направить пакет. Например, если на какой-либо порт маршрутизатора поступает пакет, адресованный в сеть N6, то из таблицы маршрутизации следует, что адрес следующего маршрутизатора — $IP_{21}$, то есть очередным этапом движения данного пакета будет движение к порту 1 маршрутизатора 2.

Поскольку пакет может быть адресован в любую сеть составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо всех сетях, входящих в составную сеть. Но при таком

подходе в случае крупной сети объем таблиц маршрутизации резко возрастает, что приводит к увеличению времени их просмотра и большего места для хранения. Поэтому на практике широко известен прием уменьшения количества записей в таблице маршрутизации, основанный на введении маршрута по умолчанию (default route), учитывающего особенности топологии сети. Рассмотрим, например, маршрутизаторы, находящиеся на периферии составной сети. В их таблицах достаточно записать номера только тех сетей, которые непосредственно подсоединены к данному маршрутизатору или расположены поблизости на тупиковых маршрутах. Обо всех остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию (default router). В примере на маршрутизаторе 4 имеются специфические маршруты только для пакетов, следующих в сети N1 – N6. Для всех остальных пакетов, адресованных в сети N7-N18, маршрутизатор предлагает продолжить путь через один и тот же порт $IP_{51}$ маршрутизатора 5, который в данном случае и является маршрутизатором по умолчанию.

## 24.2. Таблицы маршрутизации конечных узлов

Задачу маршрутизации решают не только промежуточные узлы (маршрутизаторы), но и конечные узлы — компьютеры. Решение этой задачи начинается с того, что средствами протокола IP на конечном узле определяется, направлен ли пакет в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, это означает, что пакет маршрутизировать не требуется. В противном случае маршрутизация нужна.

Структуры таблиц маршрутизации конечных узлов и транзитных маршрутизаторов аналогичны. Обратимся снова к сети, изображенной на рис. 49. Таблица маршрутизации конечного узла B, принадлежащего сети N3, могла бы выглядеть так, как табл. 7. Здесь $IP_B$ - сетевой адрес интерфейса компьютера B. На основании этой таблицы конечный узел B выбирает, на какой из двух имеющихся в локальной сети N3 маршрутизаторов (R1 или R3) следует посылать тот или иной пакет.

Еще одним отличием работы маршрутизатора и конечного узла является способ построения таблицы маршрутизации. Если маршрутизаторы, как правило, автоматически создают таблицы

маршрутизации, обмениваясь служебной информацией, то для конечных узлов таблицы маршрутизации часто создаются вручную администраторами и хранятся в файлах на дисках.

**Таблица 7** Таблица маршрутизации конечного узла B

| Номер сети назначения | Сетевой адрес следующего | Сетевой адрес | Расстояние до сети |
|---|---|---|---|
| N1 | $IP_{13}$(R1) | $IP_B$ | 1 |
| N2 | $IP_{13}$(R1) | $IP_B$ | 1 |
| N3 | — | $IP_B$ | 0 |
| N4 | $IP_{31}$ (R3) | $IP_B$ | 1 |
| N5 | $IP_{13}$(R1) | $IP_B$ | 2 |
| N6 | $IP_{31}$ (R3) | $IP_B$ | 2 |
| Маршрут по | $IP_{31}$ (R3) | $IP_B$ | — |

## 24.3. Пример IP-маршрутизации без масок

Рассмотрим процесс продвижения пакета в составной сети на примере IP-сети, показанной на рис. 50. При этом будем считать, что все узлы сети, рассматриваемой в примере, имеют адреса, основанные на классах. Особое внимание будет уделено взаимодействию протокола IP с протоколами разрешения адресов ARP и DNS.

Итак, пусть пользователю компьютера cit.mgu.com, находящегося в сети 129.13.0.0 необходимо установить связь с FTP-сервером. Пользователю известно символьное имя сервера unix.mgu.com, поэтому он набирает на клавиатуре команду обращения к FTP-серверу:

> ftp.unix.mgu.com

Выполнение этой команды инициирует три последовательные операции.

DNS-клиент (работающий на компьютере cit.mgu.com) передает DNS-серверу сообщение, в котором содержится запрос об IP-адресе сервера unix.mgu.com, с которым он должен связаться по протоколу FTP.

DNS-сервер, выполнив поиск, передает ответ DNS-клиенту о найденном IP-адресе сервера unix.mgu.com.

FTP-клиент, работающий на том же компьютере (cit.mgu.com), используя найденный адрес сервера (unix.mgu.com) передает сообщение работающему на нем FTP-серверу.

Последовательно рассмотрим, как при решении этих задач взаимодействуют между собой протоколы DNS, IP, ARP и Ethernet и

что происходит при этом с кадрами и пакетами:

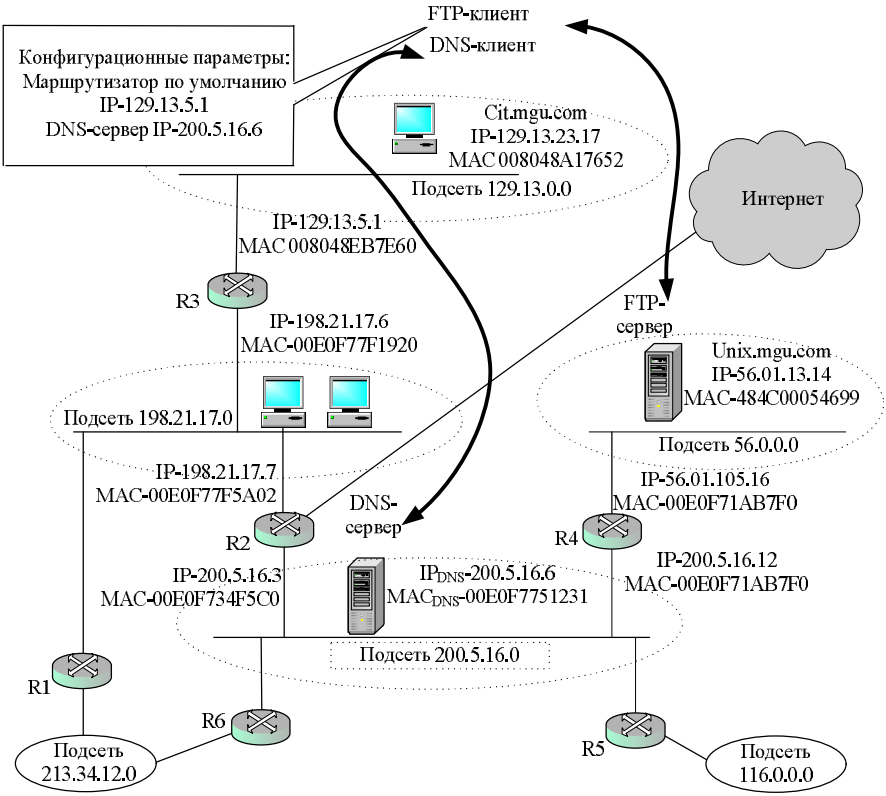



**Рис. 50** Пример IP-маршрутизации

**1. Формирование пакета с инкапсулированным в него DNS-запросом.** Программный модуль FTP-клиента, получив команду > ftp unix.mgu.com, передает запрос к работающей на этом же компьютере клиентской части протокола DNS, которая, в свою очередь формирует к DNS-серверу запрос, интерпретируемый примерно так: «Какой IP-адрес соответствует символьному имени unix.mgu.com?» Запрос упаковывается в UDP-дейтаграмму затем в IP-пакет. В заголовке пакета в качестве адреса назначения указывается IP-адрес 200.5.16.6 DNS-сервера. Этот адрес известен программному обеспечению клиентского компьютера, так как он входит в число его конфигурационных параметров. Сформированный IP-пакет будет перемещаться по сети в неизменном виде (как показано на рис. 51), пока не дойдет до адресата — DNS-сервера.

**2. Передача кадра Ethernet с IP-пакетом маршрутизатору R3.** Для передачи этого IP-пакета необходимо его упаковать в кадр Ethernet, указав в заголовке MAC-адрес получателя. Технология Ethernet способна доставлять кадры только тем адресатам, которые находятся в пределах одной подсети с отправителем. Если же адресат расположен в этой подсети, то кадр надо передать ближайшему маршрутизатору, чтобы тот взял на себя заботу о дальнейшем перемещении пакета. Для этого модуль IP, сравнив номера сетей в адресах отправителя и получателя, то есть 129.13.23.17 и 200.5.16.6, выясняет, что пакет направляется в другую сеть, следовательно, его необходимо передать маршрутизатору, в данном случае маршрутизатору по умолчанию. IP-адрес маршрутизатора по умолчанию также известен клиентскому узлу, поскольку он входит в число конфигурационных параметров. Однако для кадра Ethernet необходимо указать не IP-адрес, а MAC-адрес получателя. Эта проблема решается с помощью протокола ARP, который для ответа на вопрос: «Какой MAC-адрес соответствует IP-адресу 194.87.23.1?» делает поиск в своей ARP-таблице. Поскольку обращения к маршрутизатору происходят часто, будем считать, что нужный MAC-адрес обнаруживается в таблице и имеет значение 008048EB7E60. После получения этой информации клиентский компьютер cit.mgu.com отправляет маршрутизатору R3 пакет, упакованный в кадр Ethernet (рис. 52).


**Рис. 51** IP-пакет с инкапсулированным в него DNS-запросом

**3. Определение IP-адреса и MAC-адреса следующего маршрутизатора R2.** Кадр принимается интерфейсом 129.13.5.1 маршрутизатора R3. Протокол Ethernet, работающий на этом интерфейсе, извлекает из этого кадра IP-пакет и передает его протоколу IP. Протокол находит в заголовке пакета адрес назначения 200.5.16.6 и просматривает записи своей таблицы маршрутизации. Пусть маршрутизатор R3 не обнаруживает специфического маршрута для адреса назначения 200.5.16.6, но находит в своей таблице следующую запись:

200.5.16.0 198.21.17.7 198.21.17.6

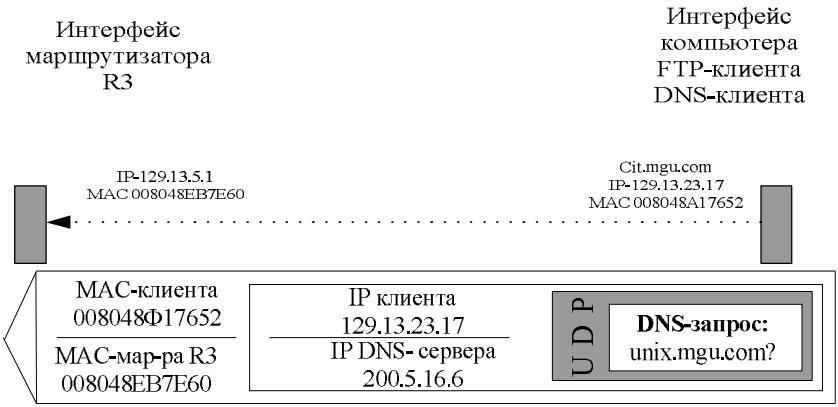



**Рис. 52** Кадр Ethernet с инкапсулированным IP-пакетом, отправленный с клиентского компьютера

Эта запись говорит о том, что пакеты для сети 200.5.16.0 маршрутизатор R3 должен передавать на свой выходной интерфейс 198.21.17.6, с которого они поступят на интерфейс следующего маршрутизатора R2, имеющего IP-адрес 198.21.17.7. Однако знание IP-адреса недостаточно, чтобы передать пакет по сети Ethernet. Необходимо определить MAC-адрес маршрутизатора R2. Как известно, такой работой занимается протокол ARP. Пусть на этот раз в ARP-таблице нет записи об адресе маршрутизатора R2. Тогда в сеть отправляется широковещательный ARP-запрос, который поступает на все интерфейсы сети 198.21.17.0. Ответ приходит только от интерфейса маршрутизатора R2 и показывает, что его IP-адрес - 198.21.17.7, MAC-адрес – 00E0F77F5A02 (рис. 53).

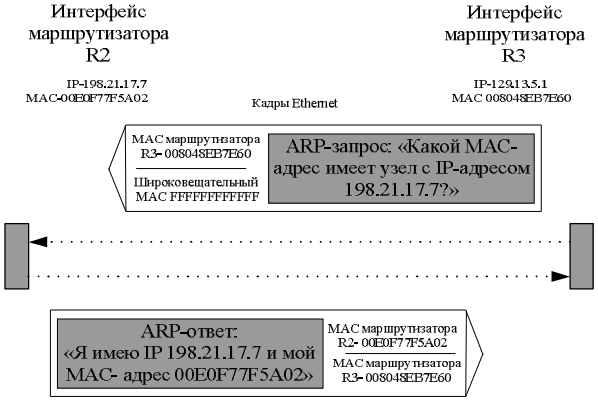



**Рис. 53** Кадры Ethernet с вложенным ARP-запросом и ARP-ответом

Теперь, зная MAC-адрес маршрутизатора R2, маршрутизатор R3 отсылает ему IP-пакет с DNS-запросом (рис. 54).

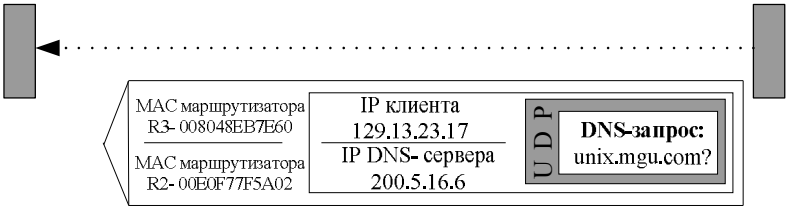



**Рис. 54** Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R3 маршрутизатору R2

**4. Маршрутизатор R2 доставляет пакет DNS-серверу.** Модуль IP на маршрутизаторе R2, отбросив заголовок кадра Ethernet, извлекает из пакета IP-адрес назначения и просматривает свою таблицу маршрутизации. Там он обнаруживает, что сеть назначения 200.5.16.0 является непосредственно присоединенной к его второму интерфейсу. Следовательно, пакет не нужно маршрутизировать, однако требуется определить MAC-адрес узла назначения (DNS-сервера). Протокол ARP «по просьбе» протокола IP находит (либо из ARP-таблицы, либо по запросу) требуемый MAC-адрес 00E0F7751231 DNS-сервера. Получив ответ о MAC-адресе, маршрутизатор R2 отправляет в сеть назначения кадр Ethernet с DNS запросом (рис. 55).

**5. Сетевой адаптер DNS-сервера захватывает кадр Ethernet, обнаруживает совпадение MAC-адреса назначения, содержащегося в заголовке, со своим собственным адресом и направляет его модулю IP.** После анализа полей заголовка IP из пакета извлекаются данные вышележащих протоколов. DNS-запрос передается программному модулю DNS-сервера. DNS-сервер просматривает свои таблицы, возможно, обращается к другим DNS-серверам и в результате формирует ответ, смысл которого состоит в следующем «Символьному имени unix.mgu.com соответствует IP-адрес 56.01.13.14».

Процесс доставки DNS-ответа клиенту cit.mgu.com совершенно аналогичен процессу передачи DNS-запроса, который был только что описан. Работая в тесной кооперации, протоколы IP, ARP и Ethernet передают клиенту DNS-ответ через всю составную сеть (рис. 56).

FTP-клиент, получив IP-адрес FTP-сервера, посылает ему свое сообщение, используя те же описанные ранее механизмы доставки данных через составную сеть. Однако будет весьма полезно мысленно воспроизвести этот процесс, обращая особое внимание на значения адресных полей заголовков кадров и заголовка вложенного IP-пакета.

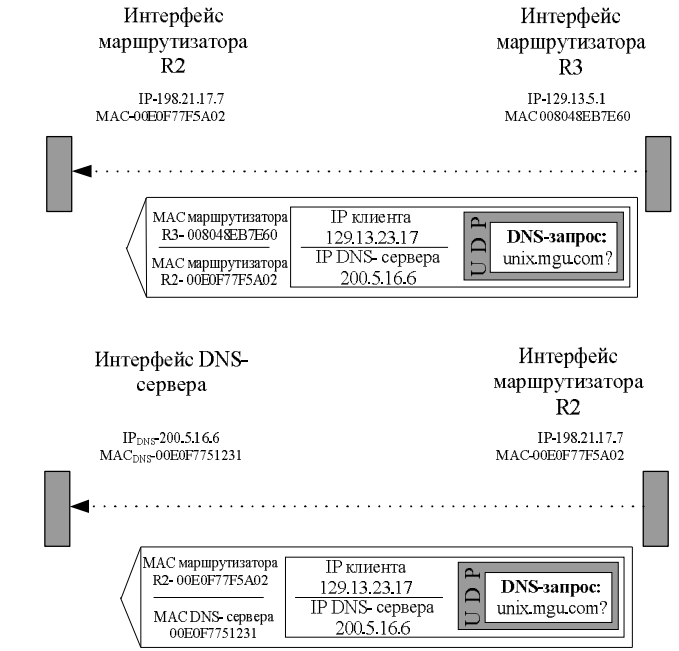



**Рис. 55** Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R2

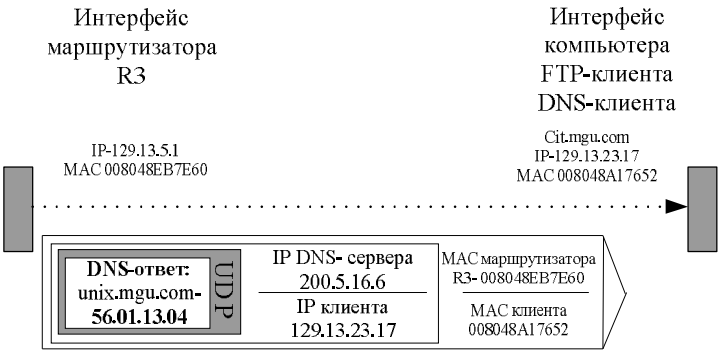



**Рис. 56** Кадр Ethernet с DNS-ответом, отправленный с маршрутизатора R3 компьютеру-клиенту

# 25. Маршрутизация с использованием масок

## 25.1. Структуризация сети масками одинаковой длины

Допустим, администратор получил в свое распоряжение сеть класса B: 129.44.0.0. Он может организовать сеть с большим числом узлов, номера которых доступны ему из диапазона 0.0.0.1-0.0.255.254. Всего в его распоряжении имеется (216 - 2) адреса. Вычитание двойки связано с учетом того, что адреса из одних нулей и одних единиц имеют специальное назначение и не годятся для адресации узлов. Однако ему не нужна одна большая неструктурированная сеть. Администратору нужно разделить сеть на три отдельных подсети, при этом трафик в каждой подсети должен быть надежно локализован. Это позволит легче диагностировать сеть и проводить в каждой из подсетей особую политику безопасности.

На рис. 57 показано разделение всего полученного администратором адресного диапазона на 4 равные части — каждая по 214 адресов. При этом число разрядов, доступных для нумерации узлов, уменьшилось на два бита, а префикс (номер) каждой из четырех сетей стал длиннее на два бита. Следовательно, каждый из четырех диапазонов можно записать в виде IP-адреса с маской, состоящей из 18 единиц, или в десятичной нотации 255.255.192.0.

129.44.0.0/18 (10000001 00101100 **00**000000 00000000)
129.44,64.0/18 (10000001 00101100 **01**000000 00000000)
129.44.128.0/18 (10000001 00101100 **10**000000 00000000)
129.44.192.0/18 (10000001 00101100 **11**000000 00000000)

Из приведенных записей видно, что администратор получает возможность использования для нумерации подсетей два дополнительных бита (выделенных жирным шрифтом). Именно это позволяет ему сделать из одной централизованно выделенной сети четыре, в данном примере это

129.44.0.0/18, 129.44.64.0/18, 129.44.128.0/18, 129.44.192.0/18.

Пример сети, построенной путем деления на 4 сети равного размера, показан на рис. 58. Весь трафик во внутреннюю сеть 129.44.0.0, направляемый из внешней сети, поступает через маршрутизатор R1. В целях структуризации информационных потоков во внутренней сети установлен дополнительный маршрутизатор R2. Каждая из вновь образованных сетей 129.44.0.0/18, 129.44.64.0/18, 129.44.128.0/18 и 129.44.192.0/18 подключена к соответственно сконфигурированным портам внутреннего маршрутизатора R2.

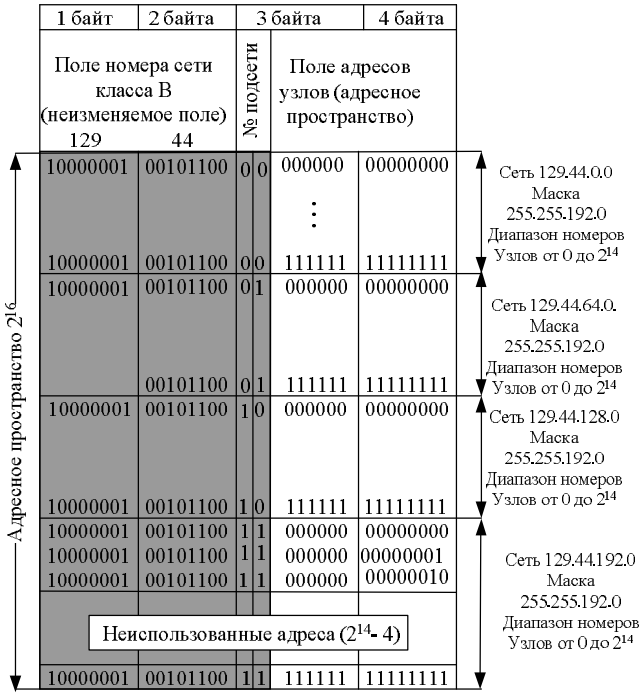



**Рис. 57** Разделение адресного пространства 129.44.0.0 сети класса B на четыре равные части

Извне сеть по-прежнему выглядит, как единая сеть класса B. Однако поступающий в сеть общий трафик разделяется локальным маршрутизатором R2 между четырьмя сетями. В условиях, когда механизм классов не действует, маршрутизатор должен иметь другое средство, которое позволило бы ему определять, какая часть 32-разрядного числа, помещенного в поле адреса назначения, является номером сети. Именно этой цели служит дополнительное поле маски, включенное в таблицу маршрутизации (табл. 8).

Первые четыре записи в таблице соответствуют внутренним подсетям, непосредственно подключенным к портам маршрутизатора R2.

Запись 0.0.0.0 с маской 0.0.0.0 соответствует маршруту по умолчанию.

Последняя запись определяет специфический маршрут к узлу 129.44.128.15. В тех строках таблицы, в которых в качестве адреса

назначения указан полный IP-адрес узла, маска имеет значение 255.255.255.255. В отличие от всех других узлов сети 129.44.128.0, к которым пакеты поступают с интерфейса 129.44.128.5 маршрутизатора R2, к данному узлу они должны приходить через маршрутизатор R3.

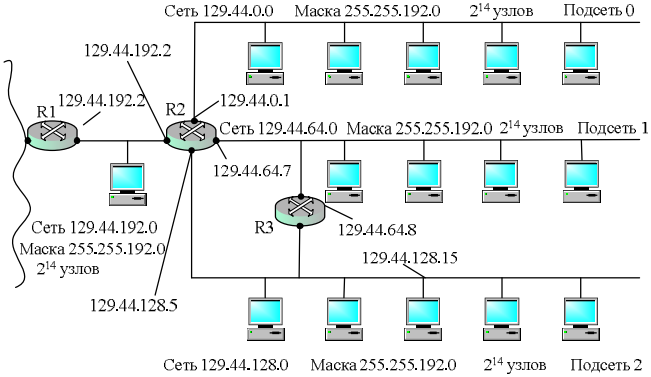



**Рис. 58** Маршрутизация с использованием масок одинаковой длины

**Таблица 8** Таблица маршрутизатора R2 в сети с масками одинаковой длины

| Адрес назначения | Маска | Адрес следующего | Адрес порта | Расстояние |
|---|---|---|---|---|
| 129.44.0.0 | 255.255.192.0 | 129.44.0.1 | 129.44.0.1 | Подключена |
| 129.44.64.0 | 255.255.192.0 | 129.4464.7 | 129.44.64.7 | Подключена |
| 129.44.128.0 | 255.255.192.0 | 129.44.128.5 | 129.44.128.5 | Подключена |
| 129.44.192.0 | 255.255.192.0 | 129.44.192.1 | 129.44.192.1 | Подключена |
| 0.0.0.0 | 0.0.0.0 | 129.44.192.2 | 129.44.192.1 | — |
| 129.44.128.15 | 255.255.255.255 | 129.44.64.8 | 129.44.64.7 | — |

## 25.2. Перекрытие адресных пространств

Рассмотрим пример использования масок для организации перекрывающихся адресных пространств.

Пусть на некотором предприятии было принято решение обратиться к поставщику услуг для получения пула адресов, достаточного для создания сети, структура, которой показана на рис. 59. Сеть клиента включает три подсети. Две из них — это надежно защищенные от внешних атак внутренние сети отделов: сеть Ethernet на 600 пользователей и сеть Token Ring на 200 пользователей.

Предприятие также предусматривает отдельную, открытую для доступа извне сеть на 10 узлов, главное назначение которой — предоставление информации в режиме открытого доступа для потенциальных клиентов. Такого рода участки корпоративной сети, в которых располагаются веб-серверы, FTP-серверы и другие источники публичной информации, называют демилитаризованной зоной (Demilitarized Zone, DMZ).

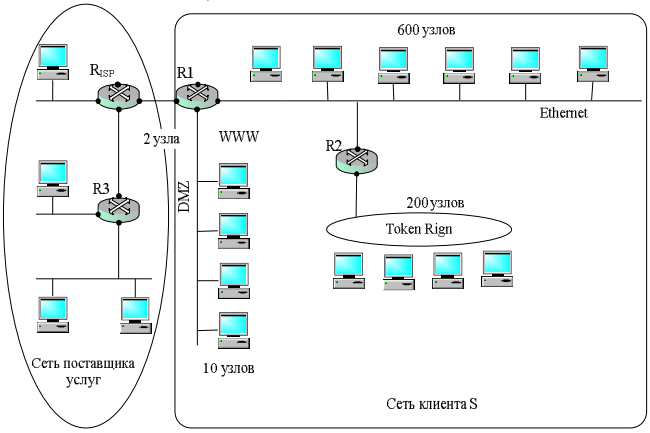



**Рис. 59** Сети поставщика услуг и клиента

Ещё одна сеть на два узла потребуется для связи с поставщиком услуг, то есть общее число адресов, требуемых для адресации сетевых интерфейсов, составляет 812. Кроме того, необходимо, чтобы пул доступных адресов включал для каждой из сетей широковещательные адреса, состоящие только из единиц, а также адреса, состоящие только из нулей. Учитывая также, что в любой сети адреса всех узлов должны иметь одинаковые префиксы, становится очевидным, что минимальное количество адресов, необходимое клиенту для построения задуманной сети, может значительно отличаться от значения 812, полученного простым суммированием.

В данном примере поставщик услуг решает выделить клиенту непрерывный пул из 1024 адресов. Значение 1024 выбрано как наиболее близкое к требуемому количеству адресов, равному степени двойки ($2^{10} = 1024$). Поставщик услуг выполняет поиск области такого размера в имеющемся у него адресном пространстве — 131.57.0.0/16, часть которого, как показано на рис. 60, уже распределена. Обозначим распределенные участки и владеющих ими клиентов через S1, S2 и S3. Поставщик услуг находит среди нераспределенных еще адресов непрерывный участок размером 1024 адреса, начальный адрес

которого кратен размеру данного участка. Таким образом, наш клиент получает пул адресов 131.57.8.0/22, обозначенный на рисунке через S.

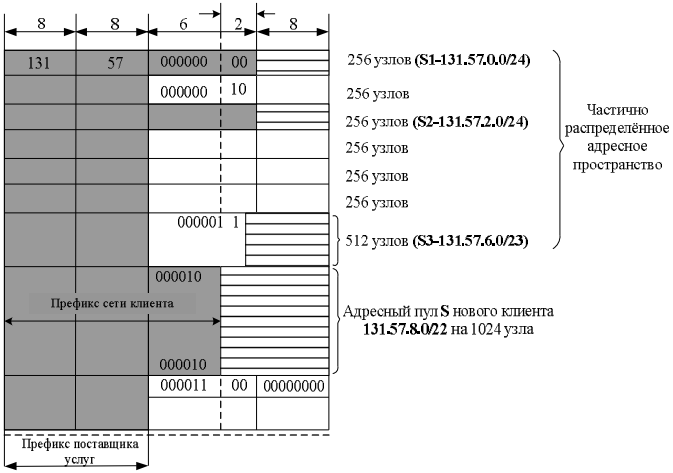



**Рис. 60** Адресное пространство поставщика услуг

Далее начинается самый сложный этап — распределение полученного от поставщика услуг адресного пула S между четырьмя сетями клиента. Прежде всего, администратор решил назначить для самой большой сети (Ethernet на 600 узлов) весь пул адресов 131.57.8.0/22, полученной от поставщика услуг (рис. 61).

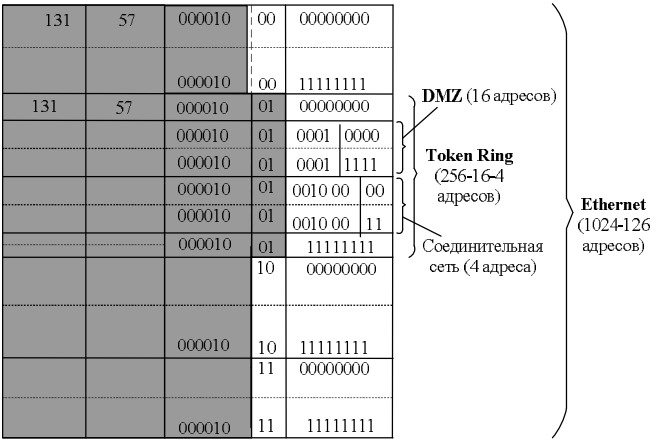



**Рис. 61** Планирование адресного пространства для сетей клиента

Номер, назначенный для этой сети, совпадает с номером сети, полученным от поставщика услуг. А как же быть с оставшимися тремя сетями? Администратор учел, что для сети Ethernet требуется только 600 адресов, а из оставшихся 624 «выкроил» сеть Token Ring 131.57.9.0/24 на 250 адресов. Воспользовавшись тем, что для Token Ring требуется только 200 адресов, он «вырезал» из нее два участка: для сети DMZ 131.57.9.16/28 на 16 адресов и для связывающей сети 131.57.9.32/30 на 4 адреса. В результате все сети клиента получили достаточное (а иногда и с избытком) количество адресов.

Следующий этап - это конфигурирование сетевых интерфейсов конечных узлов и маршрутизаторов. Каждому интерфейсу сообщается его IP-адрес и соответствующая маска. На рис. 62 показана сконфигурированная сеть клиента.

После конфигурирования сетевых интерфейсов должны быть созданы таблицы маршрутизации маршрутизаторов R1 и R2 клиента. Они могут быть сгенерированы автоматически или с участием администратора.

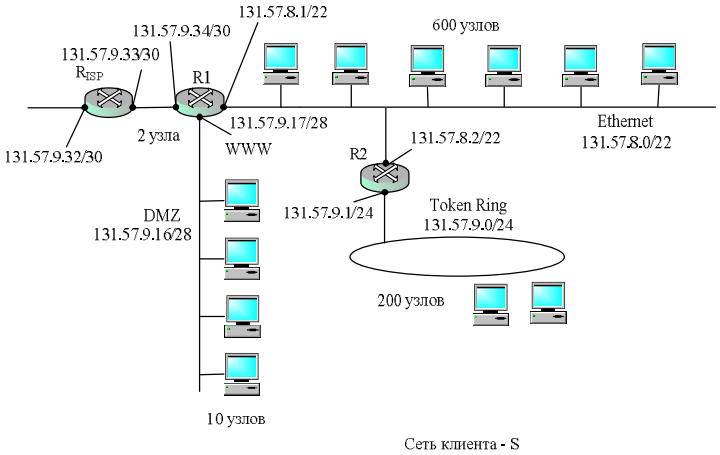



**Рис. 62** Сконфигурированная сеть клиента

## 26. Протоколы транспортного уровня TCP и UDP

К транспортному уровню стека TCP/IP относятся:

- протокол управления передачей (Transmission Control Protocol, TCP), описанный в стандарте RFC 793;
- протокол пользовательских дейтаграмм (User Datagram

Protocol, UDP), описанный в стандарте RFC 768.

Протоколы TCP и UDP, как и протоколы прикладного уровня, устанавливаются на конечных узлах.

## 26.1. Порт

Главная задача протоколов транспортного уровня TCP и UDP заключается в передаче данных между прикладными процессами, выполняющимися на компьютерах в сети.

Каждый компьютер может выполнять несколько процессов, более того, даже отдельный прикладной процесс может иметь несколько точек входа, выступающих в качестве адресов назначения для пакетов данных. Поэтому доставка данных на сетевой интерфейс компьютера-получателя — это еще не конец пути, так как данные необходимо переправить конкретному процессу-получателю. Процедура распределения протоколами TCP и UDP поступающих от сетевого уровня пакетов между прикладными процессами называется демультиплексированием (рис. 63).

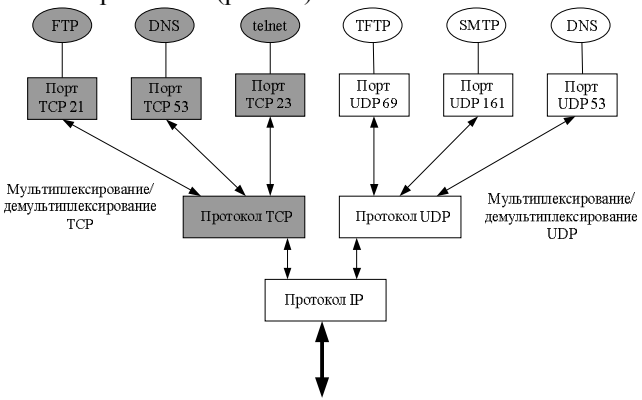



**Рис. 63** Мультиплексирование и демультиплексирование на транспортном уровне

Существует и обратная задача: данные, генерируемые разными приложениями, работающими на одном конечном узле, должны быть переданы общему для всех них протокольному модулю IP для последующей отправки в сеть. Эту работу, называемую мультиплексированием, тоже выполняют протоколы TCP и UDP.

Протоколы TCP и UDP ведут для каждого приложения две

системные очереди: очередь данных, поступающих к приложению из сети, и очередь данных, отправляемых этим приложением в сеть. Такие системные очереди называются портами, причем входная и выходная очереди одного приложения рассматриваются как один порт. Для идентификации портов им присваивают номера.

Если процессы представляют собой популярные системные службы, такие как FTP, Telnet, HTTP, TFTP, DNS и т. п., то за ними закрепляются стандартные назначенные номера, называемые также хорошо известными (well-known) номерами портов. Так, номер 21 закреплен за серверной частью службы удаленного доступа к файлам FTP, 23 — за серверной частью службы удаленного управления Telnet, 80 – за гипертекстовыми протоколами. Назначенные номера из диапазона от 0 до 1023 являются уникальными в пределах Интернета и закрепляются за приложениями централизованно.

Для тех приложений, которые еще не стали столь распространенными, номера портов назначаются локально разработчиками этих приложений или операционной системой. На каждом компьютере операционная система ведет список занятых и свободных номеров портов. При поступлении запроса от приложения, выполняемого на данном компьютере, операционная система выделяет ему первый свободный номер. Такие номера называют динамическими. В дальнейшем все сетевые приложения должны адресоваться к данному приложению с указанием назначенного ему динамического номера порта. После того как приложение завершит работу, его номер возвращается в список свободных и может быть назначен другому приложению. Динамические номера являются уникальными в пределах каждого компьютера, но при этом обычной является ситуация совпадения номеров портов приложений, выполняемых на разных компьютерах. Как правило, клиентские части известных приложений (DNS,WWW,FTP,Telnet и др.) получают динамические номера портов от ОС.

В том и другом случаях это могут быть как назначенные, так и динамические номера. Диапазоны чисел, которым выделяются номера TCP- и UDP-портов, совпадают: от 0 до 1023 для назначенных и от 1024 до 65 535 для динамических.

Стандартные назначенные номера портов уникально идентифицируют тип приложений (FTP, или HTTP, или DNS и т. д.), однако они не могут использоваться для однозначной идентификации прикладных процессов, связанных с каждым из этих типов приложений. Пусть, например, на одном хосте запущены две копии DNS-сервера — DNS-сервер 1, DNS-сервер 2 (рис. 64). Каждый из этих

DNS-серверов имеет хорошо известный UDP-порт 53. Какому из этих серверов нужно было бы направить запрос клиента, если бы в DNS-запросе в качестве идентификатора сервера был указан только номером порта?

Чтобы снять неоднозначность в идентификации приложений, разные копии связываются с разными IP-адресами. Для этого сетевой интерфейс компьютера, на котором выполняется несколько копий приложения, должен иметь соответствующее число IP-адресов – на рисунке это IP1 и IP2. Во всех IP-пакетах, направляемых DNS-серверу 1, в качестве IP-адреса указывается IP1, а DNS-серверу 2 - адрес IP2. Поэтому показанный на рисунке пакет, в поле данных которого содержится UDP-дейтаграмма с указанным номером порта 53, а в поле заголовка задан адрес IP2, буден направлен однозначно определенному адресату — DNS-серверу 2.

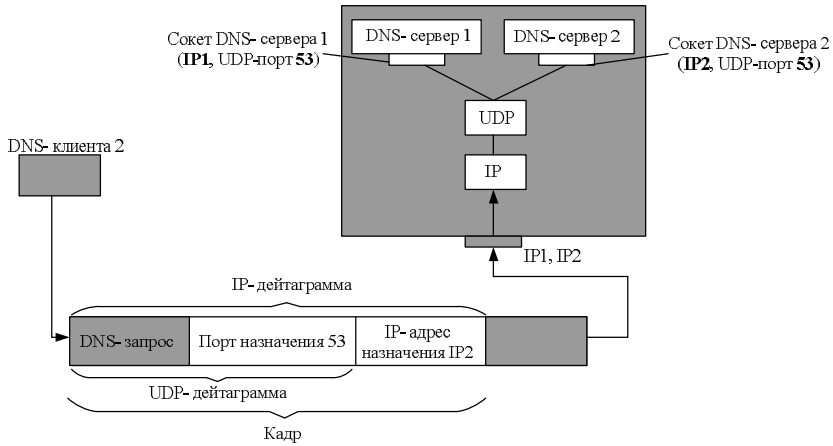



**Рис. 64** Демультиплексирование протокола UDP на основе сокетов

## 26.2. Протокол UDP и UDP-дейтаграммы

Протокол UDP, подобно IP, является дейтаграммным протоколом, реализующим так называемый надежный сервис по возможности, который не гарантирует доставку сообщений адресату.

При работе на хосте-отправителе данные от приложений поступают протоколу UDP через порт в виде сообщений (рис. 65). Протокол UDP добавляет к каждому отдельному сообщению свой 8-байтный заголовок, формируя из этих сообщений собственные

протокольные единицы, называемые UDP-дейтаграммами, и передает их нижележащему протоколу IP. В этом и заключаются его функции по мультиплексированию данных.

Каждая дейтаграмма переносит отдельное пользовательское сообщение. Сообщения могут иметь различную длину, не превышающую длину поля данных протокола IP, которая, в свою очередь, ограничена размером кадра технологии нижнего уровня. Поэтому если буфер UDP переполняется, то сообщение приложения отбрасывается.

Заголовок UDP состоит из четырех 2-байтных полей:

- номер UDP-порта отправителя;
- номер UDP-порта получателя;
- контрольная сумма;
- длина дейтаграммы.

Далее приведен пример заголовка UDP с заполненными полями:
Source Port - 0x0035
Destination Port = 0x0411
Total length = 132 (0x84) bytes
Checksum = 0x5333

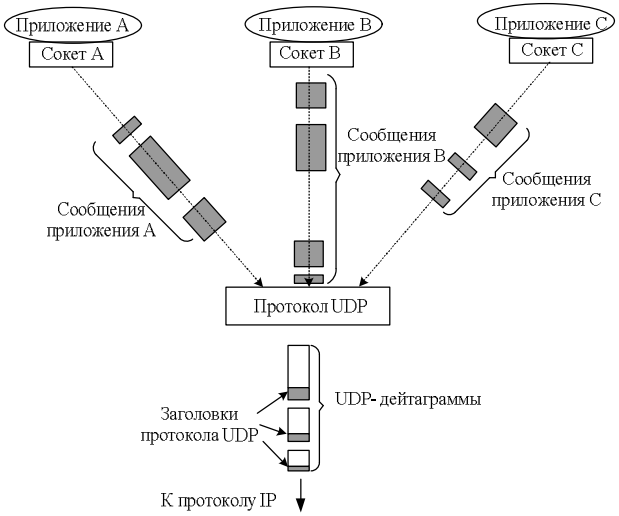



**Рис. 65** Работа протокола UDP на хосте-отправителе

В этой UDP-дейтаграмме в поле данных, длина которого, как следует из заголовка, равны (132-8) байт, помещено сообщение DNS-сервера, что можно видеть по номеру порта источника (Source Port = 0-

0035). В шестнадцатеричном формате это значение равно стандартному номеру порта DNS-сервера— 53.

Например, если контрольная сумма показывает, что в поле данных UDP-дейтаграммы произошла ошибка, протокол UDP просто отбрасывает поврежденную дейтаграмму.

Работая на хосте-получателе, протокол UDP принимает от протокола IP извлечение из пакетов UDP-дейтаграммы. Полученные из IP-заголовка IP-адрес назначения и из UDP-заголовка номер порта используются для формирования UDP-сокета, однозначно идентифицирующего приложение, которому направлены данные. Протокол UDP освобождает дейтаграмму от UDP-заголовка. Полученное в результате сообщение он передает приложению на соответствующий UDP-сокет. Таким образом, протокол UDP выполняет демультиплексирование на основе сокетов.

## 26.3. Протокол TCP и TCP-сегменты

Протокол TCP предназначен для передачи данных между приложениями. Этот протокол работает на логическом соединении, что позволяет ему обеспечивать гарантированную доставку пакета, используя в качестве инструмента ненадежный дейтаграммный сервис протокола IP.

При работе на хосте-отправителе протокол TCP рассматривает информацию, поступающую к нему от прикладных процессов, как неструктурированный поток байтов (рис. 66). Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая называется сегментом и снабжается заголовком.

**ПРИМЕЧАНИЕ**

Поясним значение однобитных полей, называемых флагами, или кодовыми битами (Code bits). Они расположены сразу за резервным полем и содержат служебную информацию о типе данного сегмента. Положительное значение сигнализируется установкой этих битов в единицу:

- URG - срочное сообщение;
- ACK - квитанция на принятый сегмент;
- PSH - запрос на отправку сообщения без ожидания заполнения буфера;
- RST - запрос на восстановление соединения;
- SYN - сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения;

- FIN - признак достижения передающей стороны последнего байта в потоке передаваемых данных.

Основная задача TCP протокола – это установление надежного логического соединения.

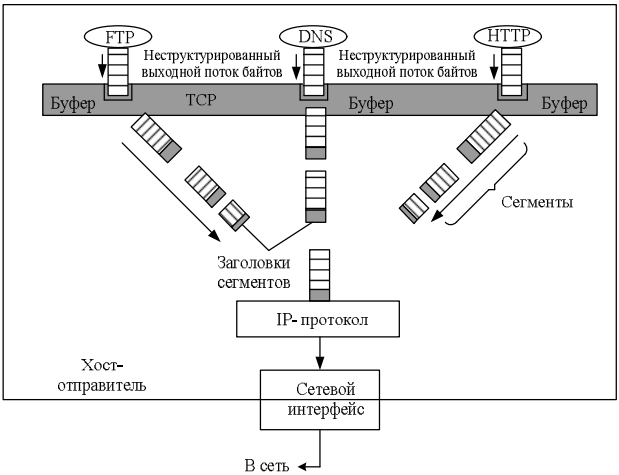



**Рис. 66** Формирование TCP-сегментов из потока байтов

| 2 байта | | | | | | | | | 2 байта |
|---|---|---|---|---|---|---|---|---|---|
| **Порт источника** (sours port) | | | | | | | | | **Порт приемника** (destination port) |
| **Последовательный номер** (sequence number) – номер первого байта в сегменте, определяет смещение сегмента относительно потока отправляемых данных | | | | | | | | | |
| **Подтвержденный номер** (acknowledgement number) - максимальный номер байта в полученном сегменте, увеличенный на единицу | | | | | | | | | |
| **Длина заголовка** (hlen) | **Резерв** (reserved) | **URG** | **ACK** | **PSH** | **RST** | **SYN** | **FIN** | | **Окно** (window) – количество байтов данных, ожидаемых отправителем данного сегмента, начиная с байта, номер которого указан в поле подтвержденного номера |
| **Контрольная сумма** (checksum) | | | | | | | | | **Указатель срочности** (urgent point) – указывает на конец данных, которые необходимо срочно принять, несмотря на переполнение буфера |
| **Параметры** (option) – это поле имеет переменную длину и может вообще отсутствовать, используется для решения вспомогательных задач, например, для согласования максимального размера сегмента | | | | | | | | | |
| **Заполнитель** (paddind) – это фиктивное поле может иметь переменную длину, используется для доведения размера заголовков до целого числа 32-битовых слов | | | | | | | | | |

**Рис. 67** Формат заголовка TCP-сегмента

# 27. Протокол RIP

Протокол RIP (Routing Information Protocol — протокол маршрутной информации) является внутренним протоколом маршрутизации дистанционно-векторного типа.

## 27.1. Построение таблицы маршрутизации

Для измерения расстояния до сети стандарты протокола RIP допускают различные виды метрик: хопы, значения пропускной способности, вносимые задержки, надежность сетей (то есть соответствующие признакам D, T и R в поле качества сервиса IP-пакета), а также любые комбинации этих метрик. Метрика должна обладать свойством аддитивности — метрика составного пути должна быть равна сумме метрик составляющих этого пути. В большинстве реализаций RIP используется простейшая метрика — количество хопов, то есть количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP на примере составной сети, изображенной на рис. 68. Мы разделим этот процесс на 5 этапов.

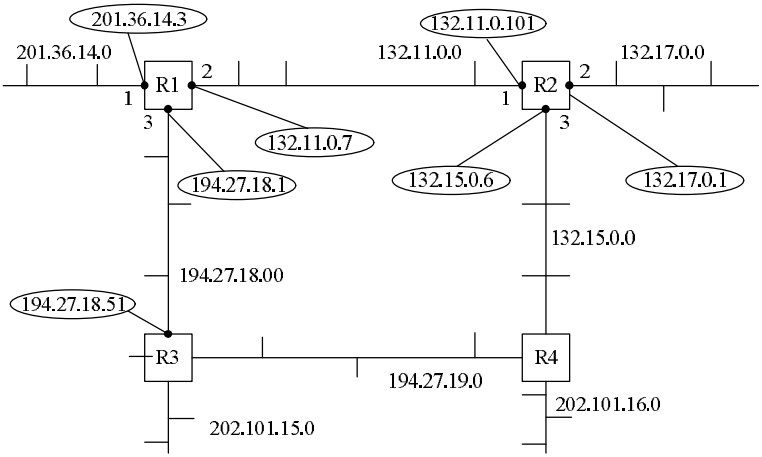



**Рис. 68** Сеть, построенная на маршрутизаторах RIP

**Этап 1 - создание минимальной таблицы.** Данная составная сеть включает восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами: Rl, R2, R3 и R4.

В исходном состоянии на каждом маршрутизаторе программным

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

Таблица 9 позволяет оценить примерный вид минимальной таблицы маршрутизации маршрутизатора R1.

**Таблица 9** Минимальная таблица маршрутизации маршрутизатора R1

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

Минимальные таблицы маршрутизации в других маршрутизаторах будут выглядеть соответственно, например, таблица маршрутизатора R2 будет состоять из трех записей (табл. 10).

**Таблица 10** Минимальная таблица маршрутизации маршрутизатора R2

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

**Этап 2 - рассылка минимальной таблицы соседям.** После инициализации каждый маршрутизатор начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица. RIP-сообщения передаются в дейтаграммах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние от маршрутизатора, который передает сообщение до сети.

Маршрутизатор R1 передает маршрутизаторам R2 и R3 следующие сообщения:

- сеть 201.36.14.0, расстояние 1;
- сеть 132.11.0.0, расстояние 1;
- сеть 194.27.18.0, расстояние 1.

**Этап 3 — получение RIP-сообщений от соседей и обработка полученной информации.** После получения аналогичных сообщений от маршрутизаторов R2 и R3 маршрутизатор наращивает каждое

полученное поле метрики на единицу и запоминает, через какой порт, и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора станет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации (табл. 11).

**Таблица 11** Таблица маршрутизации маршрутизатора R1

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

Дублирующие поля удаляются из таблицы.

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (с меньшим расстоянием в хопах), чем имеющаяся.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

**Этап 4 – рассылка новой таблицы соседям.** Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные обо всех известных ему сетях: как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

**Этап 5 – получение RIP-сообщений от соседей и обработка полученной информации.** Этап 5 повторяет этап 3 — маршрутизаторы принимают RIP-сообщения, обрабатывают находящуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации.

## Литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2005. – 864 с.

2. TCP/IP Семейство протоколов передачи данных в сетях компьютеров: Пер. с англ. / Хезер Остерлох. – СПб.: ООО «ДиаСофт ЮП», 2002. – 576 с.

3. Олифер В. Г., Олифер Н.А. Новые технологии и оборудование IP-сетей.– СПб.: Питер, 2000. – 688 с.

4. Бигелоу С. Сети: поиск неисправностей, поддержка и восстановление: Пер. с англ. – СПб.: БХВ – Петербург, 2005. – 1200 с.

5. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации – СПб.: Питер, 2005. – 703 с.

6. Кульгин М. Технологии корпоративных сетей. Энциклопедия. – СПб.: Питер, 2001.

**Учебное пособие**

**Мерзляков Сергей Александрович**
старший преподаватель

**Елизаров Дмитрий Викторович**
кандидат технических наук, доцент

# СЕТИ ЭВМ В СИСТЕМАХ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И ПРОИЗВОДСТВ

## ТЕКСТЫ ЛЕКЦИЙ

Корректор Габдурахимова Т.М.
Худ. редактор Федорова Л.Г.

Сдано в набор 03.06.2013.
Подписано в печать 27.08.2013.
Бумага писчая. Гарнитура Таймс.
Усл.печ.л. 7,2. Тираж 100.
Заказ №36.

НХТИ (филиал) ФГБОУ ВПО «КНИТУ»,
г. Нижнекамск, 423570, ул. 30 лет Победы, д. 5а.

115