Раздел 4. ГЛОБАЛЬНЫЕ СЕТИ

Совокупность различных сетей (подсетей, ЛВС), расположенных на значительных расстояниях друг от друга и объединенных в единую сеть с помощью телекоммуникационных средств, представляет собой ***территориально-распределенную сеть***, которую можно рассматривать как совокупность различных сред передачи, коммуникационных протоколов и систем управления сетями. Примерами территориально-распределенных сетей являются корпоративные сети организаций, объединяющие офисные сети, расположенные в разных городах, регионах и даже на разных континентах, городские, региональные, государственные сети и т.п.

Современные средства телекоммуникаций объединяют множество взаимосвязанных территориально-распределённых и локальных вычислительных сетей (представляющие собой подсети) различных организаций практически всего земного шара в единую сеть – ***глобальную вычислительную сеть*** Internet.

Поскольку территориально-распределённые и глобальные сети используют одинаковые принципы, технологии и оборудование, то их принято называть единым термином – **глобальные сети** или ***Wide Area Network*** (**WAN)**.

Для корректной работы глобальных сетей необходимо все сетевые стандарты связать так, чтобы они могли сосуществовать друг с другом, включая сети не на ЛВС-стандартах, такие как сети X.25 или IBM SNA.

4.1. Принципы организации глобальных сетей

4.1.1. Характерные особенности глобальных сетей

В отличие от ЛВС характерными особенностями глобальных сетей являются следующие.

1. *Неограниченный* территориальный охват.

2. Сеть объединяет ЭВМ самых *разных классов* (от персональных до суперЭВМ), локальные и территориальные сети *разных технологий*.

3. Для объединения различных сетей и передачи данных на большие расстояния используется специальное оборудование, а именно: *аппаратура передачи данных* (модемы, приемопередатчики и т.п.) и *активное сетевое оборудование* (маршрутизаторы, коммутаторы, шлюзы).

4. Топология глобальных сетей, в общем случае, *произвольная*.

5. Одной из важнейших задач, решаемой при построении глобальной сети, является организация эффективной *маршрутизации* передаваемых данных.

6. Глобальная сеть может содержать *каналы связи разных типов*: кабельные оптические и электрические, в том числе телефонные, беспроводные радио и спутниковые каналы, имеющие различные пропускные способности (от нескольких кбит/с до сотен Гбит/с).

4.1.2. Достоинства глобальных сетей

1. Предоставление пользователям сети неограниченного доступа к любым вычислительным и информационным ресурсам, а также множества специфических услуг, таких как электронная почта, голосовая связь, конференцсвязь, телевидение по запросу, доступ к разнообразным информационным ресурсам и т.д.

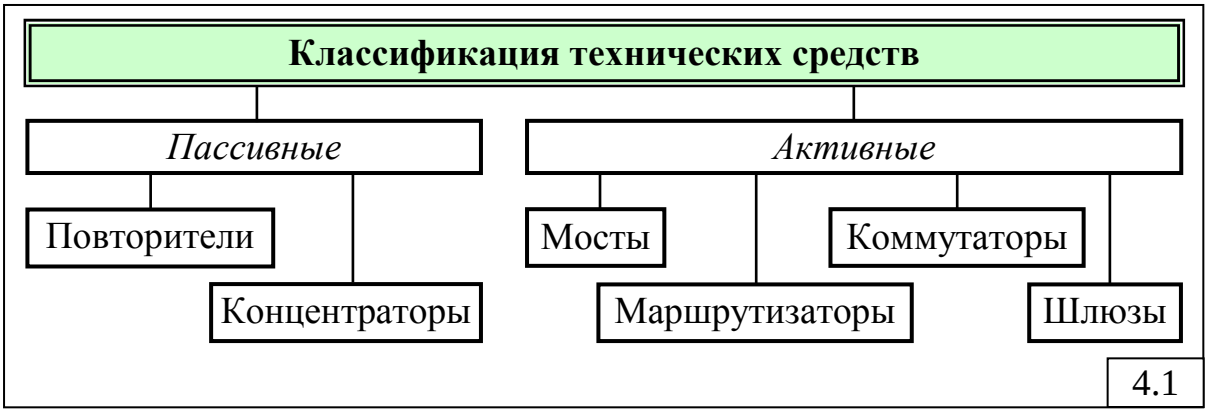
2. Возможность доступа к ресурсам сети практически из любой точки Земного шара.

3. Возможность передачи по сети любых видов данных, в том числе таких специфических как аудио и видео.

4.2. Технические средства объединения сетей

Классификация технических средств объединения сетей, представленная на рис.4.1, включает в себя:

- *пассивные* технические средства, используемые для объединения отдельных сегментов и расширения ЛВС, к которым относятся:
  - повторители (repeater);
  - концентраторы (hub);
- *активные* технические средства, используемые для построения территориально-распределённых и глобальных сетей путём объединения как ЛВС, так и сетей других не ЛВС-технологий:
  - мосты (bridg);
  - маршрутизаторы (router);
  - коммутаторы (switch);
  - шлюзы (gateway).

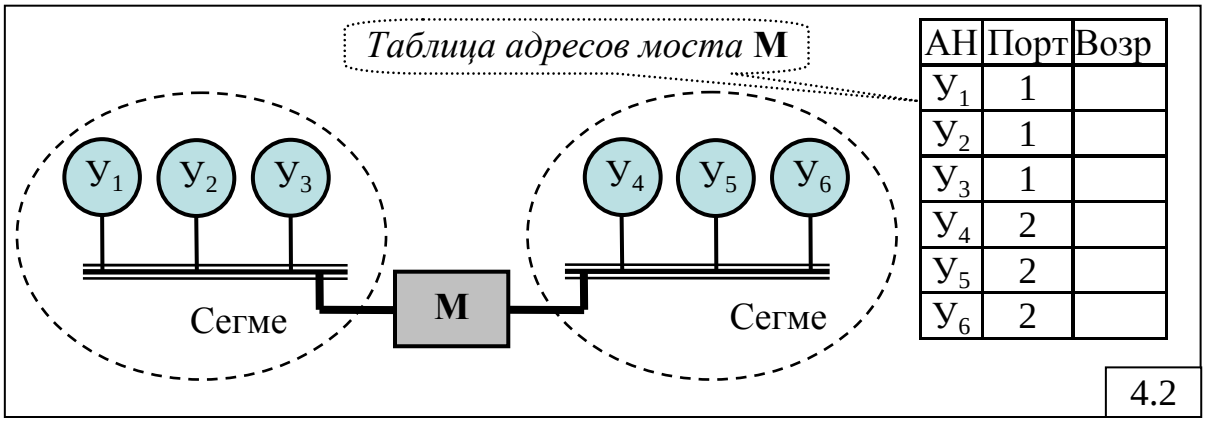


Активные технические средства, в отличие от пассивных, основной функцией которых является усиление передаваемого сигнала, управляют трафиком на основе адресов назначения передаваемых данных, то есть работают на 2-м и более высоких уровнях OSI-модели. Пассивные технические средства работают, в основном, на 1-м физическом уровне.

4.2.1. Мосты

Мост – простейшее сетевое устройство, объединяющее локальные или удаленные сегменты и регулирующее прохождение кадров между ними. Подсоединенные к мосту сегменты образуют логически единую

сеть, в которой любая станция может использовать сетевые ресурсы, как своего сегмента, так и всех доступных через мост сегментов (рис.4.2).



Мост работает на *подуровне MAC* второго канального уровня и прозрачен для протоколов более высоких уровней, то есть принимает решение о передаче кадра из одного сегмента в другой на основании физического адреса (MAC-адреса) станции назначения. Для этого мост формирует **таблицу адресов** (ТА), которая содержит (рис.4.2):

- список MAC-адресов (адресов назначения, **АН**) станций, подключенных к мосту;
- направление (**порт**), к которому станция подключена;
- "**возраст**" с момента последнего обновления этой записи.

Так как кадры, предназначенные для станции того же сегмента, не передаются через мост, трафик локализуется в пределах сегментов, что снижает нагрузку на сеть и повышает информационную безопасность. В отличие от повторителя, который действует на физическом уровне и всего лишь повторяет и восстанавливает сигналы, мост *анализирует целостность* кадров и *фильтрует кадры*, в том числе испорченные.

Мосты не нагружают работой остальные сетевые устройства – они находятся в одной большой сети с единым сетевым адресом и разными MAC-адресами.

Для получения информации о местоположении станций мосты изучают адреса станций, читая адреса всех проходящих через них кадров. При получении кадра мост сравнивает адрес назначения с адресами в ТА и, если такого адреса нет, то мост передает кадр по всем направлениям (кроме отправителя кадра). Такой процесс передачи называется "затоплением" (flooding). Если мост находит в ТА адрес назначения, то он сравнивает номер порта из ТА с номером порта, по которому пришёл кадр. Их совпадение означает, что адреса отправителя и получателя расположены в одном сегменте сети, следовательно, кадр не надо транслировать, и мост его игнорирует. Если же адреса отправителя и получателя расположены в разных сегментах, мост отправляет кадр в нужный сегмент сети.

Достоинствами мостов являются:

- относительная простота и дешевизна объединения ЛВС;

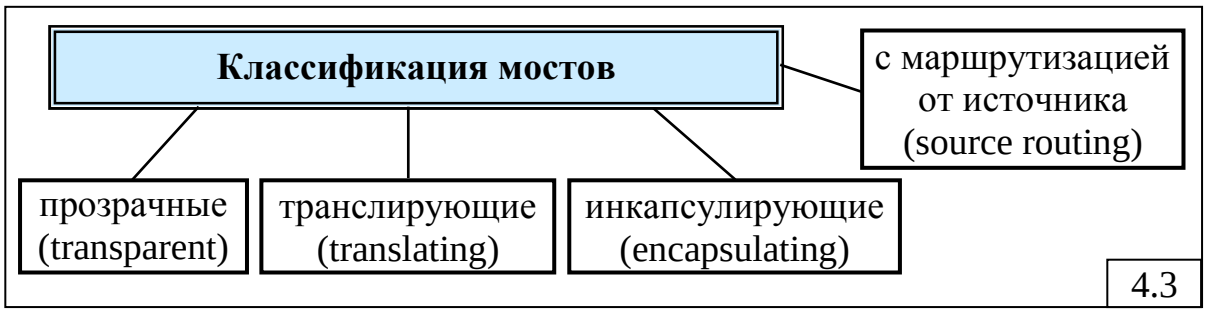
- "местные" (локальные) кадры остаются в данном сегменте и не загружают дополнительно другие сегменты;
- присутствие мостов прозрачно для пользователей;
- мосты автоматически адаптируются к изменениям конфигурации сети;
- мосты могут объединять сети, работающие с разными протоколами сетевого уровня;
- ЛВС, объединенные мостами, образуют логически единую сеть, т.е. все сегменты имеют один и тот же сетевой адрес; поэтому перемещение компьютера из одного сегмента в другой не требует изменения его сетевого адреса;
- мосты, благодаря простой архитектуре, являются недорогими устройствами.

Недостатки состоят в следующем:

- дополнительная задержка кадров в мостах;
- не могут использовать альтернативные пути; из возможных путей всегда выбирается один, остальные – блокируются;
- могут способствовать значительным всплескам трафика в сети, например, при передаче кадра, адрес которого еще не содержится в таблице моста; такие кадры передаются во все сегменты;
- не могут предотвращать "широковещательные штормы";
- не имеют средств для изоляции ошибочно функционирующих сегментов.

Существуют мосты четырех основных типов (рис.4.3):

- прозрачные (transparent);
- транслирующие (translating);
- инкапсулирующие (encapsulating);
- с маршрутизацией от источника (source routing).



4.2.1.1. Прозрачные мосты

Прозрачные мосты (transparent bridges) предназначены для объединения сетей с *идентичными протоколами* на канальном и физическом уровнях, например, Ethernet-Ethernet, Token Ring-Token Ring.

Прозрачный мост является самообучающимся устройством: в процессе работы для каждого подключенного сегмента автоматически строит таблицу адресов с адресами станций, находящихся в сегменте.

Алгоритм функционирования моста:

1) прием поступающего кадра в буфер моста;

2) анализ адреса отправителя (АО) и его поиск в таблице адресов (ТА);

3) если АО отсутствует в ТА, то этот адрес и номер порта, по которому поступил кадр, заносятся в ТА;

4) анализ адреса получателя (АП) и его поиск в ТА;

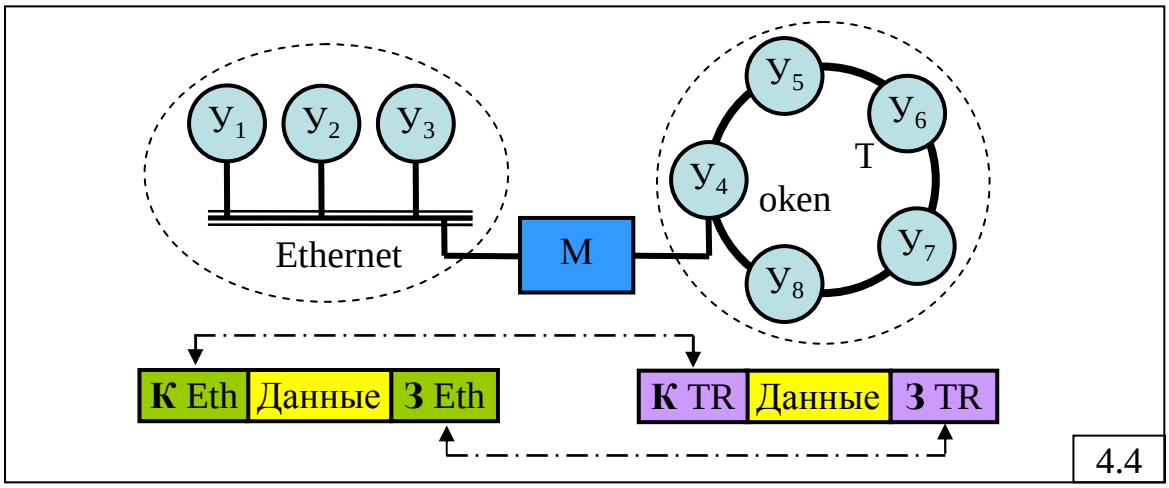
5) если АП найден в ТА, и он принадлежит тому же сегменту, что и АО (т.е. номер выходного порта совпадает с номером входного порта), кадр удаляется из буфера;

6) если АП найден в ТА, и он принадлежит другому сегменту, кадр передается в этот сегмент (на соответствующий порт);

7) если АП отсутствует в ТА, то кадр передается во все сегменты, кроме того сегмента, из которого он поступил.

4.2.1.2. Транслирующие мосты

Транслирующие мосты (translating bridges) предназначены для объединения сетей с *разными протоколами* на канальном и физическом уровнях, например, Ethernet и Token Ring (рис.4.4).

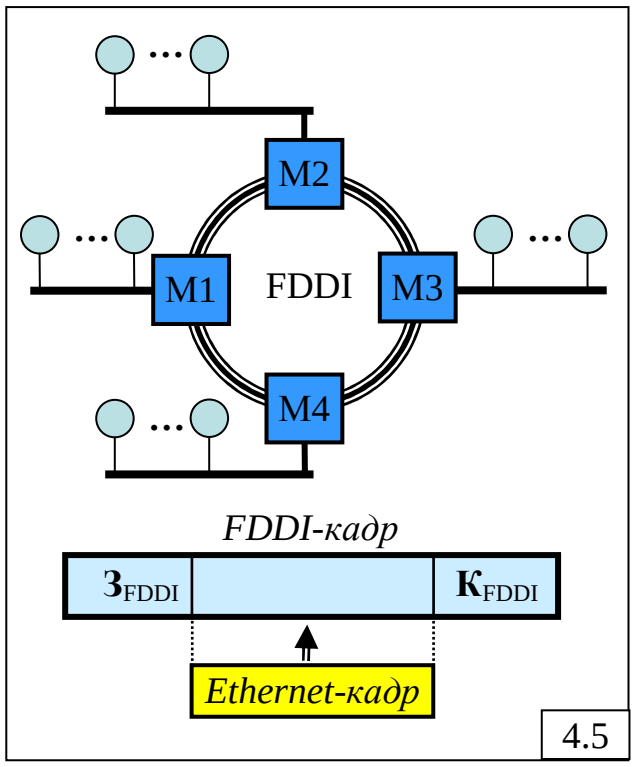


Транслирующие мосты объединяют сети путем манипулирования "конвертами": при передаче кадра из сети Ethernet в сеть TokenRing осуществляется замена заголовка (**З** ETh) и концевика (**К** Eth) Ethernet-кадра на заголовок (**З** TR) и концевик (**К** TR) TokenRing-кадра и наоборот. Поскольку в разных сетях используются кадры разной длины, а транслирующий мост не может разбивать кадры на части, то каждое сетевое устройство должно быть сконфигурировано для передачи кадров одинаковой длины.

4.2.1.3. Инкапсулирующие мосты

Инкапсулирующие мосты предназначены для объединения сетей с одинаковыми протоколами канального и физического уровня через высокоскоростную магистральную сеть с другими протоколами, например 10-мегабитные сети Ethernet, объединяемые сетью FDDI (рис.4.5).

В отличие от транслирующих мостов, которые преобразуют "конверты" одного типа в другой, инкапсулирующие мосты вкладывают полученные кадры вместе с заголовком и концевиком в другой "конверт" (см. рис.4.5), который используется в магистральной сети (отсюда термин "инкапсуляция") и передает его по этой магистрали другим мостам для доставки к узлу назначения. Конечный мост извлекает Ethernet-кадр из FDDI-кадра и передаёт его в сегмент, в котором находится адресат. Длина поля данных FDDI-кадра достаточна для размещения Ethernet-кадра максимальной длины.



4.2.1.4. Мосты с маршрутизацией от источника

Мосты с маршрутизацией от источника (source routing bridges) функционируют на основе информации, формируемой станцией, посылающей кадр, и хранимой в конверте кадра. В этом случае мостам не требуется иметь базу данных с адресами.

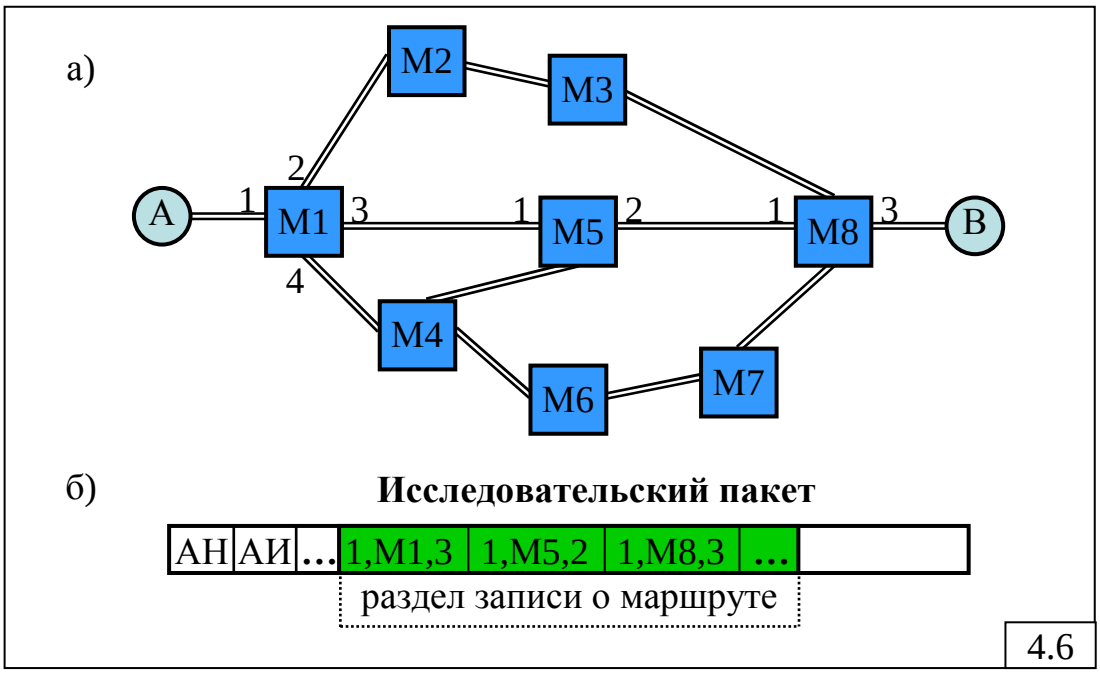
Каждое сетевое устройство определяет путь к адресату через процесс, называемый *"обнаружение маршрута"* (route discovery).

Упрощенно принцип обнаружения маршрута можно проиллюстрировать на следующем примере (рис.4.6).

Устройство-источник инициализирует обнаружение маршрута, посылая специальный кадр (рис.4.6,б), называемый *"исследовательским"* (explorer). Исследовательские кадры используют специальный конверт, распознаваемый мостами с маршрутизацией от источника. При получении такого кадра каждый мост в специально отведенное в кадре место – *поле записи о маршруте* (routing information field), заносит следующие данные: номер входного порта, с которого был получен кадр, идентификатор моста (Mi) и номер выходного порта, например: 1,M1,3 (см. рис.4.6,б). Далее мост передает этот кадр по всем направлениям, исключая то, по которому кадр был получен.

В итоге, станция назначения получает несколько исследовательских кадров, число которых определяется числом возможных маршрутов. Станция назначения выбирает один из маршрутов (самый быстрый, самый короткий или другой) и посылает ответ станции-источнику. В ответе содержится информация о маршруте, по которому должны посылаться все кадры. Станция-отправитель запоминает маршрут и использует его всегда

для отправки кадров в станцию назначения. Эти кадры при отправке вкладываются в специальные конверты, понятные для мостов с маршрутизацией от источника. Мосты, получая эти конверты, находят соответствующую запись в списке маршрутов и передают кадр по нужному направлению.

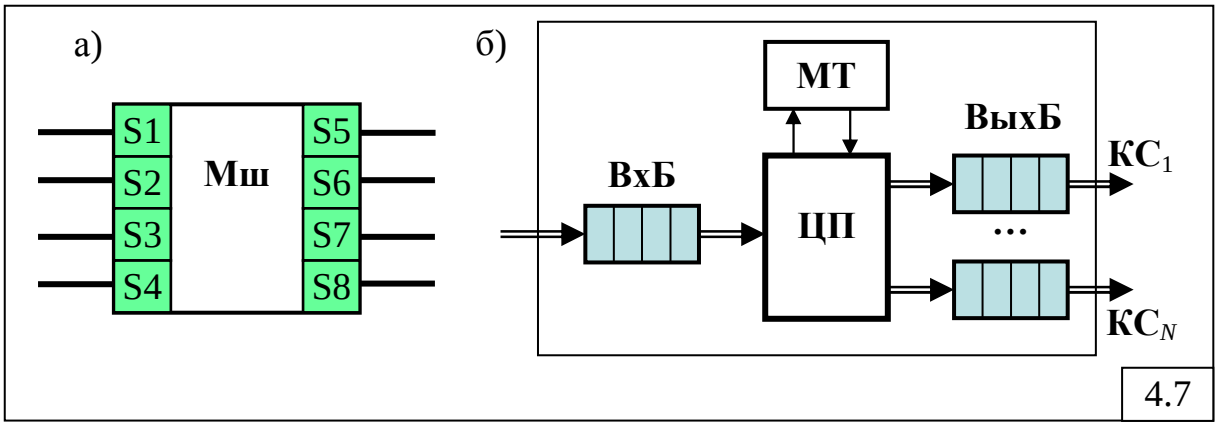


Маршрутизация от источника используется мостами в сетях Token Ring для передачи кадров между разными кольцами.

4.2.2. Маршрутизаторы

Маршрутизаторы, как и мосты, позволяют эффективно объединять сети и увеличивать их размеры, но, в отличие от последних, работают на *сетевом уровне* OSI-модели, то есть оперируют *сетевыми адресами*, и предоставляют более интеллектуальный сервис, заключающийся в определении наиболее подходящего пути и способа передачи пакетов.

В отличие от моста, работа которого прозрачна для сетевых устройств, работа маршрутизатора должна быть *явно запрошена устройством*. Для этого каждый порт (интерфейс) маршрутизатора имеет свой сетевой адрес: S1, S2, … (рис.4.7.,а). На рис.4.7,б показана каноническая структура маршрутизатора.



Поступающие пакеты заносятся во входной буфер **ВхБ**. Центральный процессор **ПМ** маршрутизатора последовательно анализирует заголовки пакетов и в соответствии с выбранной стратегией маршрутизации и заданной таблицей маршрутизации **ТМ** определяет выходной канал связи **КС**, в выходной буфер **(ВыхБ)** которого должен быть направлен пакет.

На рис.4.8 показан пример упрощённой маршрутной таблицы (МТ) узла (маршрутизатора) 4, находящегося в семиузловой сети.

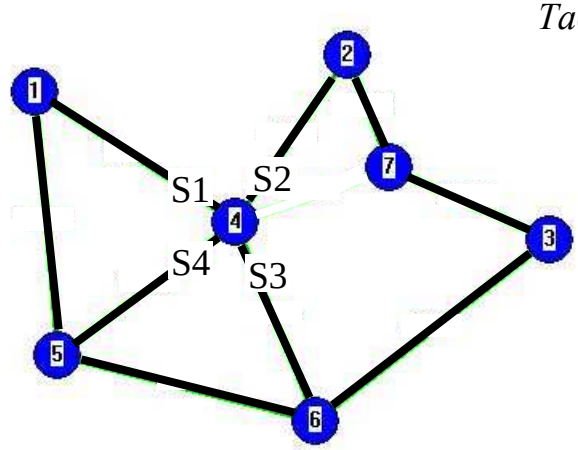


Таблица Маршрутизации (ТМ) узла 4

АН	АШ	АВИ	М
1	1	S1	0
2	2	S2	0
3	2	S2	2
3	6	S3	1
5	5	S4	0
6	6	S3	0
7	2	S2	1
default	6	S3	-

4.8

В первом столбце указаны доступные (известные) этому маршрутизатору **сетевые адреса** назначения (АН). Для каждого АН во втором столбце указывается **адрес шлюза** (АШ) – следующего маршрутизатора, к которому должны направляться пакеты, а в третьем – сетевой адрес выходного **интерфейса** (АВИ) данного маршрутизатора: S1, S2, S3, S4. При наличии альтернативных путей для одного и того же АН может быть назначено несколько возможных путей передачи пакета. Так, например, пакеты с АН=3 могут быть направлены к маршрутизатору 2 или 6 через выходные интерфейсы S2 и S3 соответственно, что отображено в таблице в виде двух строк с одним адресом назначения. В этом случае выбор маршрута осуществляется на основе **метрики** (М), указанной в 4-м столбце.

Метрика может формироваться с учётом следующих факторов:

- расстояние между источником и приемником пакета, которое обычно измеряется "счетчиками хопов" (hop – количество маршрутизаторов, пройденных пакетом от источника до приемника);
- пропускная способность канала связи;
- время доставки разными путями;
- загрузка канала связи и т.д.

В нашем примере в качестве метрики используется расстояние до адреса назначения, измеряемое в хопах.

В больших сетях для *уменьшения размера таблицы маршрутизации* и, соответственно, времени поиска маршрута, используется ограниченный набор адресов назначения, указанных в таблице явно. Для всех других

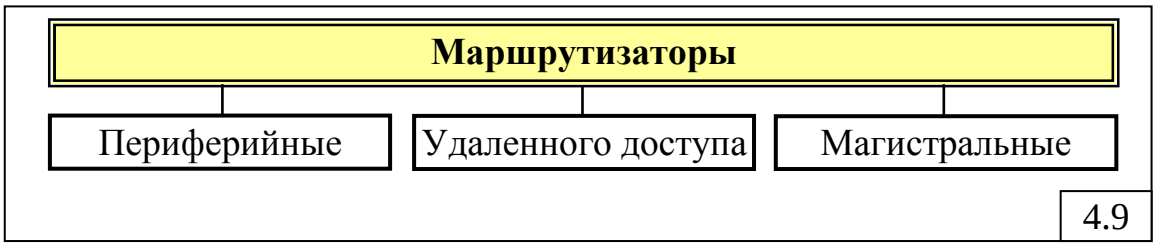
адресов используется маршрут по умолчанию, которому в таблице соответствует строка (default), указывающая соседний маршрутизатор, используемый по умолчанию.

Весь спектр маршрутизаторов можно разбить на 3 группы (рис.4.9):

1) недорогие **периферийные маршрутизаторы** для соединения небольших удаленных филиалов с сетью центрального офиса;

2) **маршрутизаторы удаленного доступа** для сетей среднего размера;

3) мощные **магистральные маршрутизаторы** для базовых сетей крупных организаций.



4.9

4.2.2.1. Периферийные маршрутизаторы

Периферийные маршрутизаторы (Boundary Router) предназначены для объединения удаленных локальных сетей с центральной сетью и, как правило, имеют ограниченные возможности: один порт для соединения с локальной сетью и один – для соединения с центральным маршрутизатором.

Все сложные функции по маршрутизации возлагаются на центральный маршрутизатор, в связи с чем периферийный маршрутизатор не требует квалифицированного обслуживания на месте и характеризуется низкой стоимостью. Основная его функция состоит в принятии решения – пересылать поступивший через порт локальной сети пакет по единственному каналу распределенной сети или нет. Тем самым исключается необходимость построения маршрутной таблицы.

4.2.2.2. Маршрутизаторы удаленного доступа

Маршрутизаторы удаленного доступа обычно имеют фиксированную (немодульную) конструкцию с небольшим числом портов, например: один LAN-порт – для сопряжения с локальной сетью, от одного до нескольких WAN-портов – для связи с маршрутизатором сети центрального офиса и один резервный порт для коммутируемого соединения.

Маршрутизаторы удаленного доступа, в общем случае, обеспечивают:

- *предоставление канала связи по требованию* (dial-on-demand) – автоматическое установление коммутируемого соединения только во время передачи данных;
- *сжатие данных*, позволяющее примерно вдвое повысить пропускную способность канала связи;

- *автоматическое переключение трафика на коммутируемые линии* (полностью или частично) в случае выхода из строя выделенных линий, а также при пиковых нагрузках.

4.2.2.3. Магистральные маршрутизаторы

Магистральные маршрутизаторы, в зависимости от архитектуры, делятся на маршрутизаторы:

- с централизованной архитектурой;
- с распределённой архитектурой.

Характерные особенности магистральных маршрутизаторов **с распределенной архитектурой**:

1) модульная конструкция:

- каждый модуль маршрутизатора снабжен собственным процессором, обрабатывающим локальный трафик, проходящий через порты этого модуля;
- центральный процессор задействуется только для маршрутизации пакетов между разными модулями;

2) наличие до нескольких десятков портов для сопряжения с локальными и территориальными сетями разных типов: Ethernet, Token Ring, FDDI, X.25, Frame Relay, ATM и т.д.;

3) поддержка средств обеспечения отказоустойчивости, необходимых для стратегически важных приложений:

- замена модулей в "горячем" режиме (без выключения питания);
- использование избыточных источников питания;
- автоматическая динамическая реконфигурация в случае отказов;
- распределенное управление.

В маршрутизаторах **с централизованной архитектурой** вся вычислительная мощность сосредоточена в одном модуле.

Основное *преимущество* магистральных маршрутизаторов с распределенной архитектурой по сравнению с централизованной – более высокие показатели производительности и отказоустойчивости.

Наиболее известными фирмами-поставщиками маршрутизаторов являются Cisco, 3Com, Hewlett-Packard.

4.2.2.4. Методы маршрутизации

Все методы маршрутизации, применяемые в маршрутизаторах, можно разбить на две группы (рис.4.10):

1) методы *статической (фиксированной)* маршрутизации;

2) методы *динамической (адаптивной)* маршрутизации.

4.10

Статическая маршрутизация означает, что пакеты передаются по определенному пути, установленному администратором и не изменяемому в течение длительного времени.

Статическая маршрутизация применяется в небольших мало изменяющихся сетях, благодаря таким *достоинствам как*:

- низкие требования к маршрутизатору;
- повышенная безопасность сети.

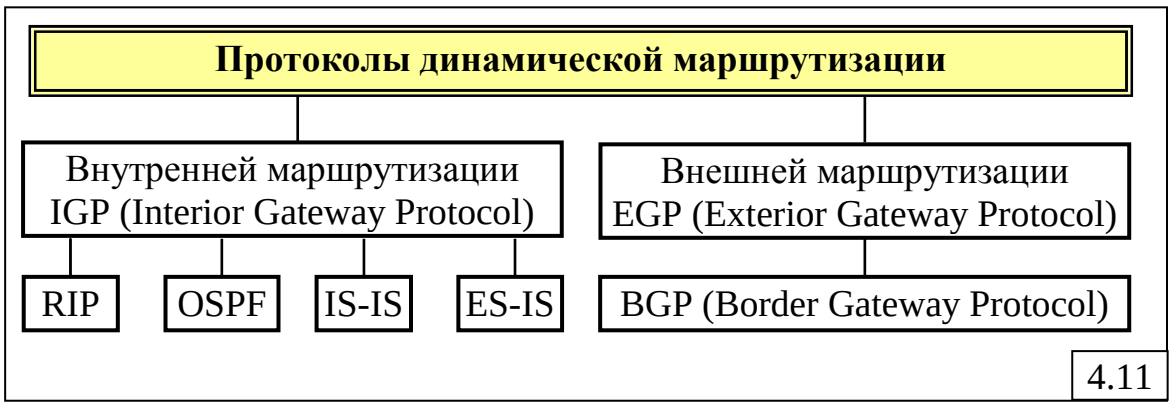
В то же время статической маршрутизации присущи следующие *недостатки*, существенно ограничивающие её применение:

- высокая трудоемкость эксплуатации (сетевые администраторы должны задавать и модифицировать маршруты вручную);
- медленная адаптация к изменениям топологии сети.

Динамическая маршрутизация – распределенная маршрутизация, позволяющая автоматически изменять маршрут следования пакетов при отказах или перегрузках каналов связи.

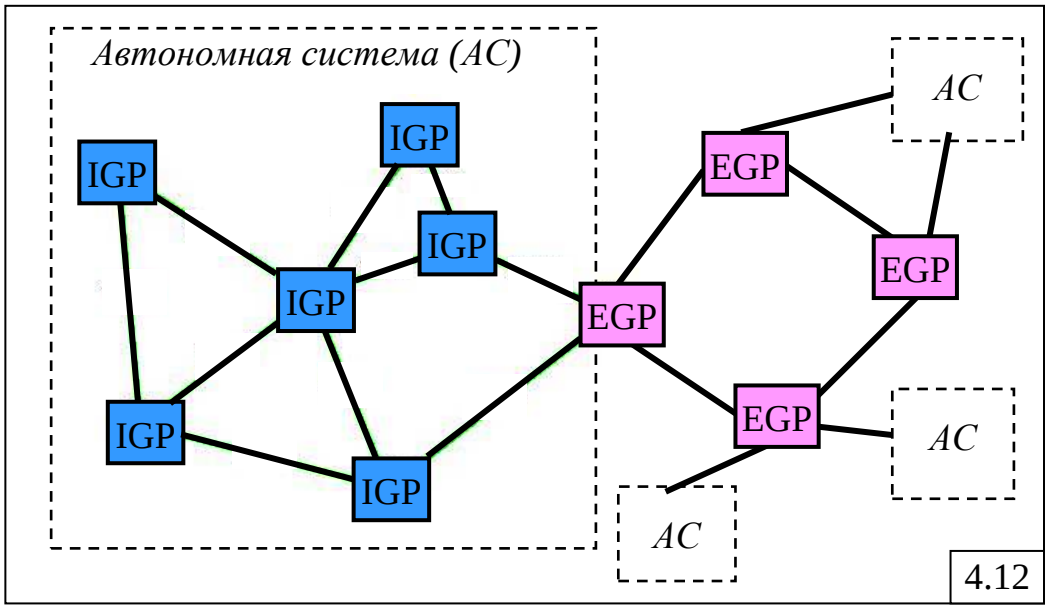
Для автоматического построения и модификации маршрутных таблиц используются протоколы (рис.4.11):

- *внутренней маршрутизации* – IGP (Interior Gateway Protocol), например RIP, OSPF, IS-IS, ES-IS;
- *внешней маршрутизации* – EGP (Exterior Gateway Protocol), например протокол BGP (Border Gateway Protocol), используемый в сети Internet.



С использованием **протоколов внутренней маршрутизации** маршрутные таблицы строятся в пределах так называемой ***автономной системы*** (autonomous system), представляющей собой совокупность сетей с единым административным подчинением (рис.4.12).

Для обмена маршрутной информацией между автономными системами чаще всего применяется **протокол внешней маршрутизации** EGP, разработанный для сети Internet. Этот протокол назван так потому, что внешний маршрутизатор, как правило, размещается на периферии автономной системы. Его задача заключается в сборе информации о доступности всех сетей данной автономной системы и последующей передаче этой информации внешним маршрутизаторам других автономных систем.



С учетом опыта применения протокола EGP был разработан протокол BGP, основанный на использовании надежного транспортного протокола TCP, который по сравнению с EGP:

- обеспечивает более быструю стабилизацию оптимальных маршрутов;
- меньше загружает сеть служебной информацией, в частности, за счет передачи при изменении сети информации, относящейся только к этому изменению.

4.2.2.5. Протоколы маршрутизации

Протоколы маршрутизации управляют динамическим обменом информацией о маршрутах между всеми маршрутизаторами сети, реализуются программно в маршрутизаторе, создавая таблицы маршрутизации, отображающие организацию всей сети.

Протоколы внутренней маршрутизации, как правило, основаны на алгоритмах обмена:

- таблицами "вектор-длина" – DVA (Distance Vector Algorithm) – протоколы типа «distance vector»;
- информацией о состоянии каналов – LSA (Link-State Algorithm) – протоколы типа «link state».

DVA - алгоритм обмена информацией о доступных сетях и расстояниях до них путём *периодической* рассылки маршрутизаторами широковещательных пакетов. К протоколам типа DVA относится один из самых первых протоколов RIP (Routing Information Protocol), который первоначально широко применялся в сети Интернет. Эти протоколы характеризуются тем, что периодически (даже если в сети не происходит изменений) посылают широковещательные пакеты с таблицами маршрутизации, которые, проходя через маршрутизаторы, обновляют таблицы маршрутизации.

В каждой строке маршрутной таблицы указываются:

- сетевой адрес некоторой сети;
- адрес маршрутизатора, через который следует передавать пакеты, направляемые в данную сеть;
- расстояние до сети.

При инициализации маршрутизатора в таблицу маршрутизации записываются:

- адреса соседних сетей;
- адреса соседних маршрутизаторов, с которыми данный маршрутизатор связан непосредственно;
- расстояние до соседних маршрутизаторов принимается равным 0 или 1 в зависимости от реализации.

Каждые 30 секунд маршрутизатор передает широковещательный пакет, содержащий пары (V, D), где V – адрес доступной сети, называемый ***вектором***, а D – расстояние до этой сети, называемое ***длиной вектора***.

В метрике RIP длина вектора измеряется *числом транзитных маршрутизаторов* (в хопах) между данным маршрутизатором и соответствующей сетью. На основании полученных таблиц "вектор-длина" маршрутизатор вносит дополнения и изменения в свою маршрутную таблицу, определяя пути минимальной длины во все доступные сети.

Поскольку каналы связи могут иметь разные пропускные способности, в некоторых реализациях RIP длина вектора умножается на весовой коэффициент, зависящий от скорости передачи данных по КС.

Основное *достоинство* RIP и других протоколов типа DVA – *простота* реализации.

Недостатки RIP:

1) *медленная стабилизация* оптимальных маршрутов;

2) *большая загрузка сети* передаваемыми таблицами "вектор-длина", обусловленная двумя основными факторами:

- периодичностью передачи широковещательных пакетов, содержащих таблицы "вектор-длина" – пакеты передаются даже в том случае, если в сети не было никаких изменений;
- большим объёмом этих таблиц, который пропорционален числу подсетей, входящих в сеть.

Протоколы типа distance vector целесообразно применять в небольших и относительно устойчивых сетях. В больших сетях периодически посылаемые широковещательные пакеты приводят к перегрузке сети и уменьшению пропускной способности.

LSA – алгоритмы обмена информацией о состоянии каналов, называемые также ***алгоритмами предпочтения кратчайшего пути SPF*** (Shortest Path First), основаны на динамическом построении маршрутизаторами карты топологии сети за счет сбора информации обо всех объединяющих их каналах связи. Для этого маршрутизатор периодически тестирует состояние каналов с соседними маршрутизаторами, помечая каждый канал как "активный" или "неактивный". На практике для уменьшения слишком частой смены этих двух состояний применяется следующее правило: «*канал считается "активным" до тех пор, пока значительный процент тестов не даст отрицательного результата, и "неактивным" – пока значительный процент тестов не даст положительного результата*».

При изменении состояния своих каналов маршрутизатор немедленно распространяет соответствующую информацию по сети всем остальным маршрутизаторам, которые, получив сообщения, обновляют свои карты сети и заново вычисляют кратчайшие пути во все точки назначения.

Достоинства алгоритмов LSA:

1) гарантированная и более быстрая стабилизация оптимальных маршрутов, чем в алгоритмах DVA;

2) простота отладки и меньший объем передаваемой информации, не зависящий от общего числа подсетей в сети.

Протоколы типа LSA используются в больших или быстро растущих сетях. К ним относятся такие протоколы, как Open Shortest Path First (OSPF) и Intermediate System to Intermediate System (IS-IS).

Самой распространенной реализацией алгоритма LSA является протокол OSPF – открытый стандарт, разработанный для применения в маршрутизаторах сети Интернет и широко используемый в настоящее время в других сетях (NetWare, SNA, XNS, DECNet).

Обладая всеми преимуществами алгоритмов LSA, протокол OSPF обеспечивает следующие дополнительные возможности.

1. *Маршрутизация* пакетов в соответствии *с заказанным типом обслуживания*. Сетевой администратор может присваивать межсетевым каналам различные значения "стоимости", основываясь на их пропускной способности, задержках или эксплуатационных расходах. Маршрутизатор выбирает путь следования пакета в результате анализа не только адреса получателя, но и поля "тип обслуживания" в заголовке.

2. *Равномерное распределение нагрузки* между альтернативными путями одинаковой стоимости (в отличие от протокола RIP, вычисляющего только один путь в каждую точку назначения).

3. *Маршрутизация* пакетов в соответствии *с классом обслуживания*. Сетевой администратор может создать несколько очередей с различными приоритетами. Пакет помещается в очередь на отправку по результатам анализа типа протокола. Для пакетов, чувствительных к временны*м* задержкам, выделяется очередь с более высоким приоритетом.

4. *Аутентификация маршрутов*. Отсутствие этой возможности, например в протоколе RIP, может привести к перехвату пакетов злоумышленником, который будет передавать таблицы "вектор-длина" с указанной малой длиной путей от своего ПК во все подсети.

5. *Создание виртуального канала между маршрутизаторами*, соединенными не напрямую, а через некоторую транзитную сеть.

В модели OSI на основе алгоритма LSA определены протоколы маршрутизации сетевого уровня:

- "оконечная система – транзитная система", ES-IS (End System-to-Intermediate System);
- "транзитная система – транзитная система", IS-IS (Intermediate System-to-Intermediate System).

Протоколы типа LSA, в отличие от DVA, посылают информацию о маршрутах только для отображения изменений в своих сетевых соединениях.

Другое отличие заключается в возможности выбора канала передачи из нескольких возможных с учетом одного из параметров маршрутизации, задаваемого пользователем:

- задержки или скорости передачи данных;
- пропускной способности или производительности;
- надежности.

Достоинства маршрутизаторов по сравнению с мостами:

- высокая безопасность данных;
- высокая надежность сетей за счет альтернативных путей;
- эффективное распределение нагрузки по каналам связи за счет выбора наилучших маршрутов для передачи данных;
- б*о*льшая гибкость за счёт выбора маршрута в соответствии с метрикой, учитывающей его стоимость, пропускную способность каналов связи и т.д.;
- гарантированная защита от "широковещательного шторма";
- возможность объединения сетей с разной длиной пакетов.

Недостатки маршрутизаторов:

- вносят сравнительно большую задержку в передачу пакетов;
- более сложны в установке и конфигурировании, чем мосты;
- при перемещении компьютера из одной подсети в другую требуется изменить его сетевой адрес;
- более дорогие, чем мосты, так как требуются более мощные процессоры, больший объем оперативной памяти, более дорогое

программное обеспечение, стоимость которого зависит от числа поддерживаемых протоколов.

В табл.4.1 сведены характерные особенности мостов и маршрутизаторов.

Таблица 4.1

Характерные особенности мостов и маршрутизаторов

Признак	*Мосты*	*Маршрутизаторы*
1. Адресация	Работают с MAC-адресами	Работают с сетевыми адресами
2. Данные	Используют только адреса отправителя и получателя	Используют много разных источников для выбора маршрута
3. Конверт	Не имеют доступа к данным в конверте	Могут разбивать пакеты на более короткие
4. Пересылка	Пакеты только отфильтровываются	Пересылают пакеты на конкретный адрес
5. Приоритеты	Не учитывают	Учитывают, обеспечивая разные типы сервиса
6. Время задержки	Небольшое, однако при перегрузках возможны потери кадров	Большая задержку, но имеют более высокую производительность
7.Надежность	Нет гарантии доставки кадров	Гарантируют доставку пакетов
8.Отказоустой-чивость	Перестают работать при неисправных сетях	Более устойчивы к отказам сети (за счет многих путей)
9.Безопасность	Могут ограничить доступ к устройствам	Обеспечивают более высокую степень защиты

Сети с протоколами, не обладающими сетевым уровнем и, соответственно, не имеющие сетевого адреса, не могут использовать маршрутизаторы и объединяются с помощью мостов или коммутаторов. Однако существуют маршрутизаторы, которые одновременно могут выполнять функции моста и называются ***мостами/маршрутизаторами*** (bridge/router или иногда brouter).

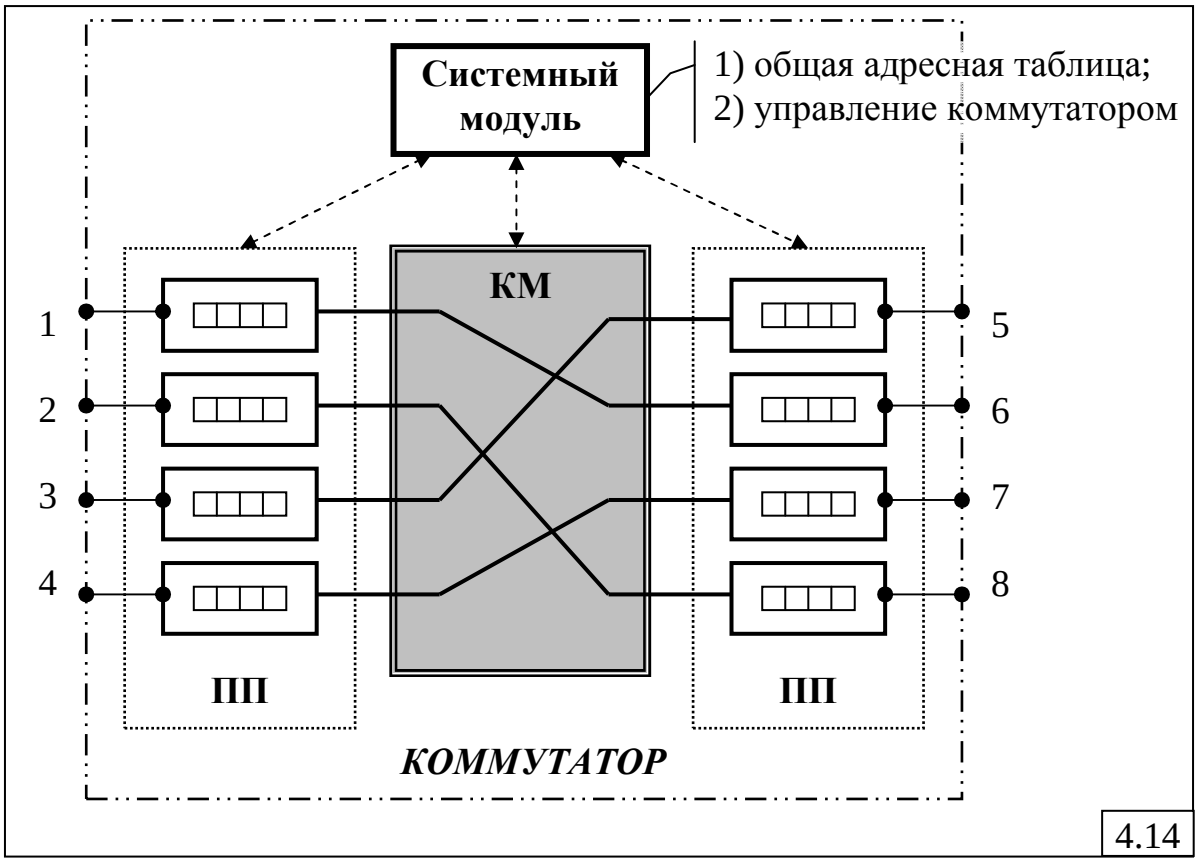
4.2.3. Коммутаторы

Технология коммутации сегментов для ЛВС Ethernet появилась в 1990 году. Коммутатор по функциональным возможностям занимает промежуточное положение между мостом и маршрутизатором и при объединении сегментов локальных сетей работает на 2-м канальном уровне, то есть коммутирует данные на основе анализа MAC-адресов.

Производительность коммутаторов значительно выше, чем мостов, и достигает нескольких миллионов кадров в секунду.

4.2.3.1. Каноническая структура коммутатора

Каноническая структура коммутатора представлена на рис.4.14.



Здесь: КМ – коммутационная матрица; ПП – процессоры портов с буферной памятью для хранения кадров.

В отличие от моста в коммутаторе каждый порт имеет свой процессор, в то время как все порты моста управляются одним процессором. В коммутаторе устанавливается один путь для всех кадров одного и того же сообщения, имеющих один адрес назначения и образующих так называемую «пачку», в то время как в маршрутизаторе для каждого пакета определяется свой наилучший путь. Передача кадров из входных буферов разных портов в выходные буферы коммутатора может происходить параллельно и независимо друг от друга. Эти особенности коммутатора обусловливает меньшие задержки при передаче данных по сравнению с маршрутизаторами.

Коммутационная матрица передаёт кадры из входных буферов в выходные на основе *таблицы коммутации*. Общее управление коммутатором и коммутационной матрицей реализуется *системным модулем*, который кроме того поддерживает общую адресную таблицу.

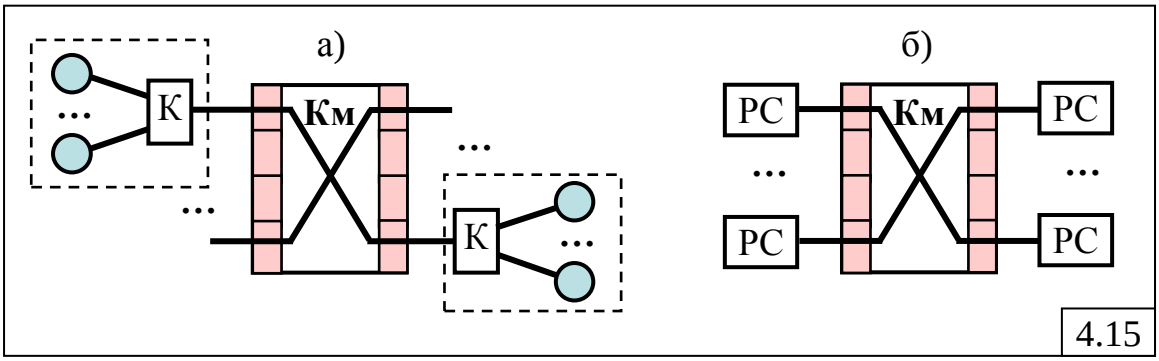
Коммутаторы могут реализовать один из двух способов коммутации:

- *с полной буферизацией кадра*, когда анализ заголовка поступающего кадра начинается только после того, как кадр будет полностью принят во входной буфер;

- *«на лету» (on-the-fly)*, когда анализ заголовка поступающего кадра начинается сразу же после того, как во входной буфер принят заголовок кадра, не ожидая завершения приёма целиком всего кадра, что позволяет ещё больше сократить задержку кадра в коммутаторе.

Коммутаторы локальных сетей могут работать в одном из двух режимов:

- ***полудуплексный***, когда к порту коммутатора подключается сегмент сети на коаксиальном кабеле или концентратор с подключенными к нему рабочими станциями (рис.4.15,а);
- ***дуплексный***, когда к каждому порту коммутатора подключается только одна рабочая станция (рис.4.15,б).



Подключение к портам коммутатора *по одной рабочей станции* (а не сегментов) называется ***микросегментацией***.

Переход на дуплексный режим требует изменения логики работы MAC-узлов и драйверов сетевых адаптеров (не фиксировать коллизии в ЛВС Ethernet, не ждать маркера в Token Ring и FDDI).

Соединения «коммутатор-коммутатор» могут поддерживать дуплексный режим.

При работе коммутатора может возникнуть ситуация, когда на один и тот же выходной порт коммутатора кадры поступают от нескольких входных портов с суммарной интенсивностью, превышающей предельное значение для данной технологии ЛВС, например для ЛВС Ethernet с пропускной способностью 10 Мбит/с – 14 880 кадров в секунду. Это приводит к перегрузкам и потерям кадров за счет переполнения выходного буфера соответствующего порта.

Для устранения подобных ситуаций необходим **механизм управления потоками кадров**.

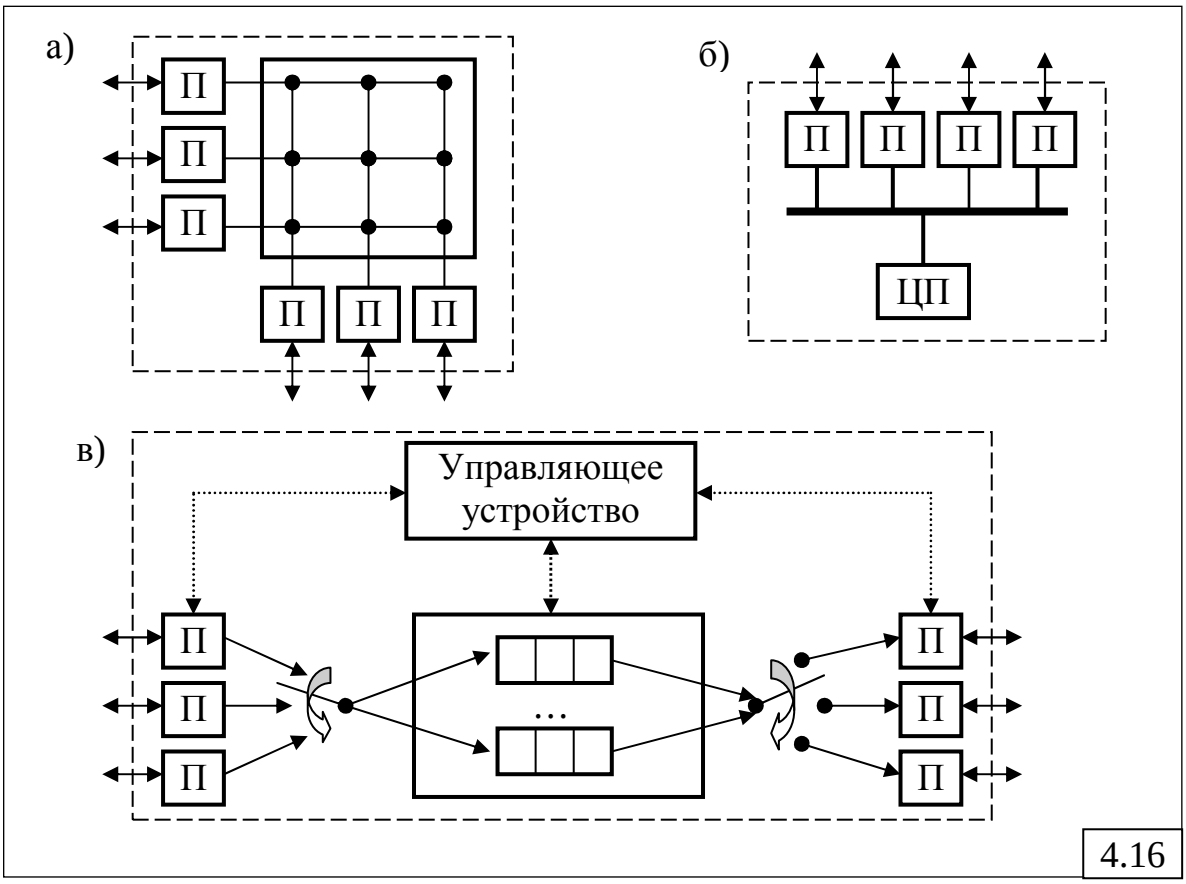
Для ЛВС Ethernet и Fast Ethernet в 1997 году принят стандарт IEEE 802.3x на управление потоком в *дуплексном режиме*, предусматривающий две команды – «Приостановить передачу» и «Возобновить передачу», которые направляются соседнему узлу. Для высокоскоростных сетей (Gigabit Ethernet и др.) с целью не допустить блокировок всех коммутаторов в сети разрабатываются более тонкие механизмы, которые указывают, *на какую величину* нужно уменьшить поток кадров, а не приостанавливать его до нуля.

При полудуплексном режиме коммутатор воздействует на конечный узел с помощью *механизмов доступа к среде*, а именно:

- ***метод обратного давления***, заключающийся в создании искусственных коллизий в сегменте с помощью jam-последовательности;
- ***метод агрессивного поведения***, когда порт коммутатора уменьшает межкадровый интервал или паузу после коллизии, что обеспечивает коммутатору преимущественный доступ к среде передачи.

4.2.3.2. Техническая реализация коммутаторов

На рис.4.16 представлены типовые варианты технической реализации коммутаторов, которые во многом повторяют варианты реализации многопроцессорных вычислительных комплексов.



Вариант 1. На основе коммутационной матрицы (рис.4.16,а).

Достоинства:

- максимальная производительность;
- высокая надежность.

Недостатки:

- сложность и высокая стоимость;
- ограниченное число портов, поскольку с их увеличением существенно возрастает стоимость.

Вариант 2. На основе общей шины (рис.4.16,б).

Достоинства:

- простота;
- дешевизна.

Недостатки:

- низкая производительность;
- низкая надежность.

Вариант 3. На основе разделяемой многовходовой памяти (рис.4.16,в). Этот вариант занимает промежуточное положение между вариантами на основе коммутационной матрицы и на основе общей шины.

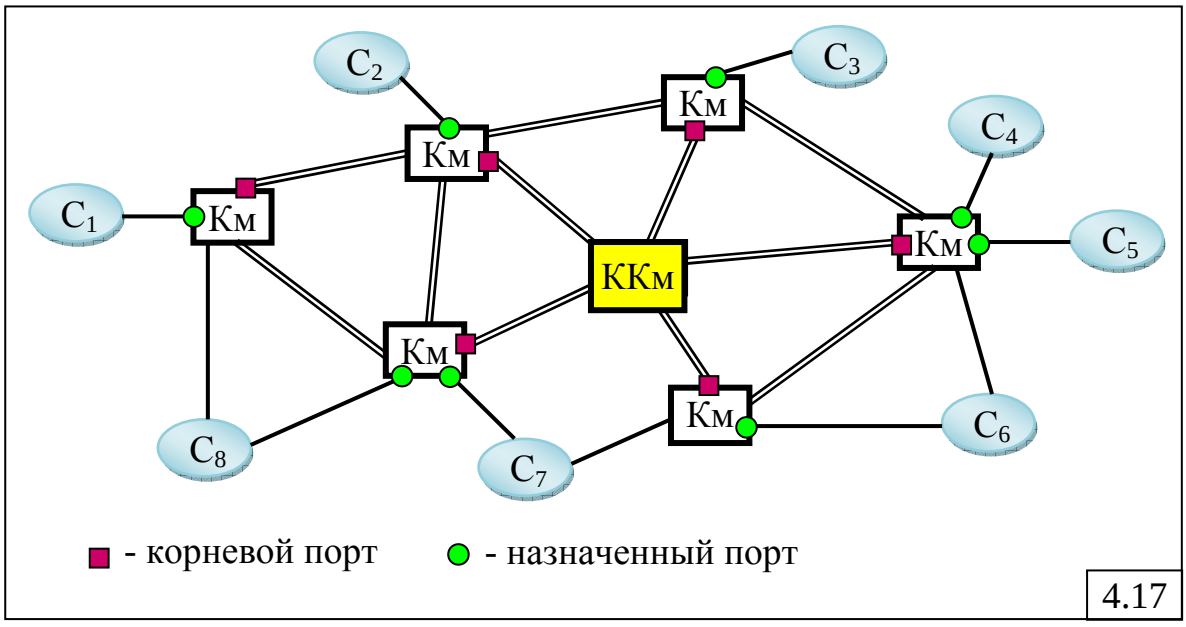
4.2.3.3. Дополнительные функции коммутаторов

Коммутаторы по сравнению с мостами являются более интеллектуальными сетевыми устройствами и обладают рядом дополнительных функций.

1. *Поддержка «алгоритма покрывающего дерева» («Spanning Tree»)*, который позволяет автоматически определять древовидную конфигурацию связей в сети для исключения петель и циклов в маршрутах (замкнутых маршрутов).

Алгоритм «Spanning Tree» реализуется в 3 этапа (рис.4.17):

- определяется автоматически (коммутатор с меньшим MAC-адресом блока управления) или назначается администратором **корневой коммутатор (ККм)**, от которого строится дерево;
- для каждого коммутатора (Км) определяется **корневой порт**, через который лежит кратчайший путь к корневому коммутатору;
- для каждого сегмента (C_i) сети выбирается **назначенный порт** – порт, который обеспечивает кратчайшее расстояние от данного сегмента до корневого коммутатора.



4.17

2. *Трансляция протоколов канального уровня.*

Коммутаторы транслируют протоколы по тем же алгоритмам, что и транслирующие мосты (в соответствии со спецификациями IEEE 802.1H и RFC 1042).

3. *Фильтрация кадров* в соответствии с заданными условиями (например, ограничивают доступ к некоторым службам сети).

4. *Приоритезация трафика* независимо от технологии сети, например путём:

- приписывания приоритета портам коммутатора;
- назначения приоритета кадрам в соответствии со стандартом IEEE 802.1p, который предусматривает общий дополнительный заголовок для кадров Ethernet, состоящий из двух байт (перед полем данных кадра), где 3 бита задают приоритет кадра.

Свойства коммутаторов, позволяющие локализовать и контролировать потоки данных, а также управлять ими с помощью пользовательских фильтров, позволяют использовать коммутаторы для построения виртуальных ЛВС (ВЛВС, VLAN – Virtual LAN).

4.2.4. Шлюзы

Шлюз – программно-аппаратный комплекс, соединяющий разнородные сети или сетевые устройства и позволяющий решать проблемы, связанные с различием протоколов и систем адресации.

Шлюзы переводят различные сетевые протоколы и позволяют различным сетевым устройствам не просто соединяться, а работать как единая сеть. В качестве примеров можно назвать пакетные адаптеры (PAD), конверторы протоколов и устройства, соединяющие сети Ethernet и X.25. В сети Internet шлюзом часто называется межсетевой маршрутизатор.

Шлюзы обеспечивают еще более интеллектуальный и более медленный сервис, чем мосты и маршрутизаторы и могут работать на высших уровнях OSI-модели.

4.3. Сети с установлением соединений

Как указывалось в разделе 1, коммутация пакетов в компьютерных сетях может быть реализована двумя способами:

- на основе *дейтаграммной* передачи пакетов *без установления соединения* между взаимодействующими абонентами сети;
- на основе *виртуального канала с установлением соединения*.

Передача пакетов на основе *виртуальных каналов* широко применяется при построении глобальных вычислительных сетей с коммутацией пакетов и обеспечивает наибольшую эффективность для долговременных устойчивых потоков данных. Передача пакетов на основе виртуальных каналов реализована в сетях X.25, Frame Relay и ATM.

4.3.1.Принцип передачи пакетов на основе виртуальных каналов

Существуют два типа виртуальных каналов:

- ***коммутируемый виртуальный канал*** (Switched Virtual Circuit, SVC), который создаётся по запросу абонента до начала передачи данных и только на время сеанса;
- ***постоянный виртуальный канал*** (Permanent Virtual Circuit, PVC), который создается вручную администратором сети (возможно, с

привлечением централизованных средств управления сетью) и не изменяется в течение достаточно длительного (в пределе неограниченного) времени.

При создании **коммутируемого виртуального канала** маршрутизация пакетов в узлах сети выполняется с использованием маршрутных таблиц *только один раз* на этапе установления соединения. При этом каждому виртуальному каналу присваивается ***идентификатор (номер) виртуального канала*** (Virtual Channel Identifier, VCI), на основе которого в дальнейшем происходит передача пакетов между узлами сети. Значение VCI имеет не глобальный характер, а локальный – действует только в пределах данного узла, причём VCI в разных узлах одного и того же виртуального канала в общем случае различны. В процессе создания виртуального канала для каждого порта узла формируются *таблицы коммутации*, предписывающие, на какой порт нужно передать пришедший пакет с определенным значением VCI. После создания виртуального канала узлы продвигают пакеты на основании значений VCI небольшой разрядности (не более 24 бит), а не адресов длиной десятки и даже сотни бит. Кроме того таблицы коммутации портов обычно содержат меньше записей, чем таблицы маршрутизации, так как хранят сведения только о действующих в данный момент соединениях, проходящих через данный порт коммутатора.

Такая организация передачи данных позволяет уменьшить задержку пакетов в сети за счет следующих факторов:

1) *решение о продвижении пакета принимается быстрее* из-за меньшего размера таблицы коммутации;

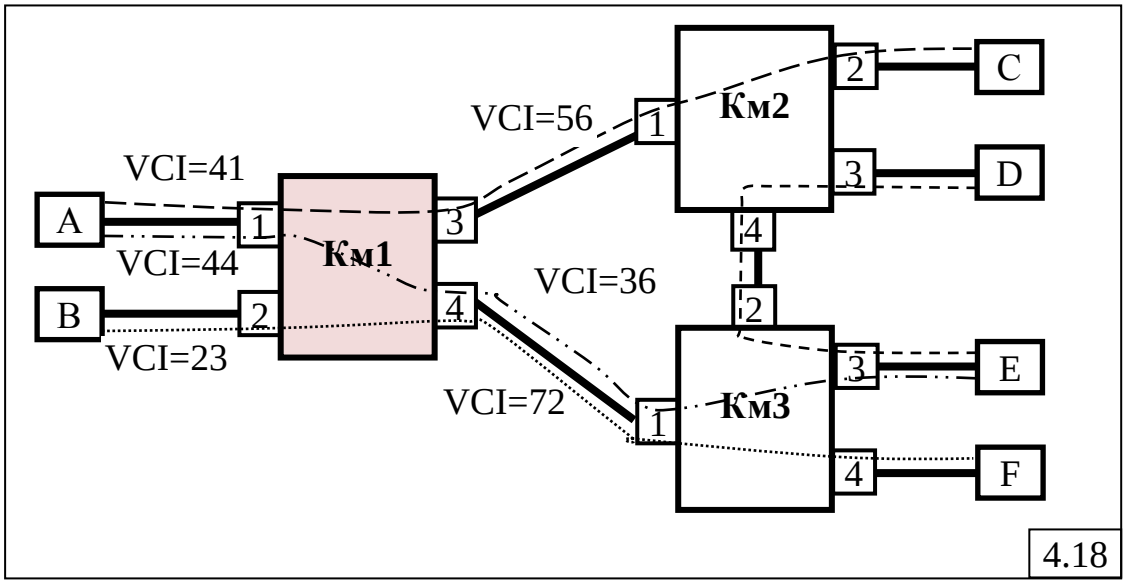
2) *возрастает эффективная (полезная) скорость передачи данных* за счет уменьшения доли служебной информации в заголовке пакета, так как идентификатор виртуального канала в заголовке пакета обычно занимает не более 24 бит, в то время как адреса конечных узлов в территориально-распределенных и глобальных сетях обычно имеют достаточно большую длину и занимают 6 и более байт.

Использование **постоянных виртуальных каналов** (PVC) более эффективно, чем коммутируемых, поскольку отсутствует этап установления соединения, и продвижение кадров выполняется на основе заранее сформированных таблиц коммутации. *Постоянный виртуальный канал* подобен *выделенному каналу* – обмен пакетами может происходить в любой момент времени. В то же время PVC отличается от выделенного канала тем, что пользователь делит пропускную способность сети с другими пользователями. С одной стороны, это обусловливает основной недостаток PVC по сравнению с выделенным каналом – отсутствие гарантий относительно реально предоставляемой пропускной способности, а с другой стороны – делает использование PVC дешевле, чем аренда выделенной линии.

Режим продвижения пакетов на основе таблицы коммутации называется ***коммутацией***, а узлы сети – ***коммутаторами***, которые

обычно работают не на третьем (сетевом), а на втором (канальном) уровне OSI-модели.

Принцип передачи пакетов на основе виртуальных каналов рассмотрим на примере фрагмента сети, представленного на рис.4.18. Узлы (компьютеры пользователей) A, B, C, D, E, F связаны в сеть с помощью 3-х четырёхпортовых коммутаторов Км1, Км2 и Км3.



Для установления соединения между конечными узлами узел-источник посылает специальный пакет – запрос на установление соединения (Call Request), который содержит адрес узла назначения и номер виртуального соединения VCI. Этот номер имеет локальное значение для каждого порта (узла и коммутатора) и выбирается из множества свободных в данный момент номеров. Через один порт можно установить достаточно большое количество виртуальных соединений.

Пусть конечный узел A, устанавливающий виртуальное соединение с узлом B, сформировал пакет Call Request на установление соединения, в котором указаны адрес назначения АН=B и номер виртуального соединения VCI=41. Пакет Call Request направляется в порт 1 коммутатора Км1 сети, где по адресу назначения с использованием таблицы маршрутизации (рис.4.19,а) определяется номера порта Км1, на который нужно переслать пакет.

В соответствии с таблицей маршрутизации пакет Call Request с порта 1 направляется в порт 3. Одновременно коммутатор заменяет в пакете номер виртуального соединения VCI=41 на новое значение, которое выбирается из множества свободных номеров для выходного порта 3. В нашем примере это значение VCI=56. Наличие разных номеров VCI для разных портов коммутатора (на входе и выходе) позволяет реализовать дуплексный режим передачи данных.

Кроме таблицы маршрутизации для каждого порта формируется таблица коммутации. В таблице коммутации входного порта 1 коммутатор отмечает, что в дальнейшем пакеты, прибывшие на этот порт с номером VCI=41 должны передаваться на порт 3, причем номер виртуального

канала должен быть изменен на 56 (рис.4.19,б). Одновременно делается запись в таблице коммутации порта 3: пакеты, поступившие с VCI=56 нужно передавать на порт 1, меняя номер виртуального канала на VCI=41. Таким образом, при получении пакетов в обратном направлении узел-источник А получает пакеты с тем же номером VCI, с которым он отправлял их к узлу В.

а)

Таблица маршрутизации

АН	**Порт**
A	1
B	2
C	3
D	3
E	4
F	4

б)

Таблицы коммутации портов

Порт 1

Вх.	**Вых.**	
VCI	Порт	VCI
41	3	56
44	4	36

Порт 2

Вх.	**Вых.**	
VCI	Порт	VCI
23	4	72

Порт 3

Вх.	**Вых.**	
VCI	Порт	VCI
56	1	41

Порт 4

Вх.	**Вых.**	
VCI	Порт	VCI
36	1	44
72	2	23

4.19

Аналогичные действия по запросу Call Request выполняются в коммутаторе Км2, где также в процессе маршрутизации формируются таблицы коммутации. После того, как пакет Call Request благополучно достигнет узла-назначения В, последний сформирует служебный пакет подтверждения, который будет передан узлу А по сформированному виртуальному каналу. Получение пакета подтверждения инициирует в узле А передачу пакетов с данными, которые будут передаваться в сети по сформированному виртуальному пути, причём значения VCI будут изменяться при передаче пакета от входного порта коммутаторов к выходному в соответствии с таблицами коммутаций по номерам виртуального соединения.

Пакеты данных уже не содержат длинные адреса конечных узлов, а имеют в заголовке только номер виртуального канала, на основании которого и производится коммутация всех пакетов, кроме пакета запроса на установление соединения. Созданный виртуальный канал не изменяется в течение всего времени существования соединения.

Таким образом, передача данных на основе виртуального канала реализуется в два этапа:

- этап маршрутизации всего одного пакета – запроса на установку виртуального соединения в соответствии с адресом назначения, в процессе которого формируются таблицы коммутации;
- этап коммутации пакетов на основании номера виртуального канала с использованием таблиц коммутации.

Использование виртуальных каналов оказывается эффективным при передаче через сеть долговременных потоков данных, но неэффективным для кратковременных потоков, так как на установление соединения уходит достаточно много времени.

4.3.2. Сети X.25

4.3.2.1. Назначение и структура сетей X.25

Стандарт X.25 «Интерфейс между оконечным оборудованием данных и аппаратурой передачи данных для терминалов, работающих в пакетном режиме в сетях передачи данных общего пользования», принятый в 1976 году и дополненный в 1984 году, наилучшим образом подходит *для передачи трафика низкой интенсивности*, характерного для терминалов, и в меньшей степени соответствует более высоким требованиям трафика локальных сетей.

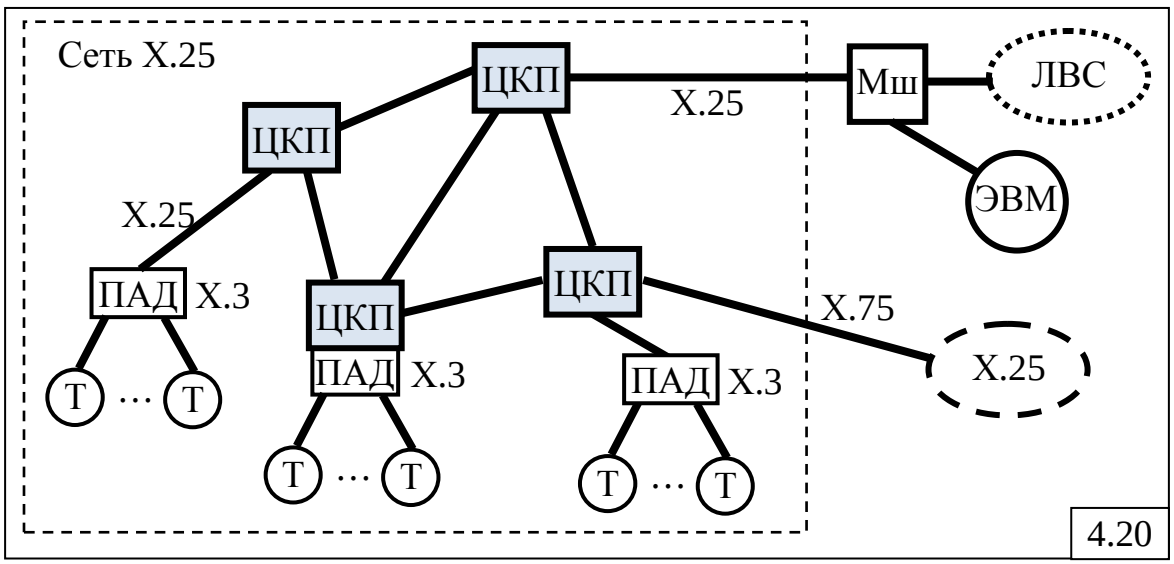
Сети, доступ к которым производится в соответствии с рекомендациями X.25, называют ***сетями X.25*** или ***сетями пакетной коммутации***. Как видно из названия, стандарт не описывает внутреннее устройство сети X.25, а только определяет пользовательский интерфейс с сетью.

Сети X.25 долгое время были единственно доступными сетями, которые хорошо работают на ненадежных линиях благодаря протоколам с установлением соединения и коррекцией ошибок на двух уровнях – канальном и сетевом.

Взаимодействие двух сетей X.25 определяет стандарт X.75.

Сети X.25 характеризуются следующими особенностями.

Сеть X.25 состоит из коммутаторов, называемых *центрами коммутации пакетов (ЦКП)*, расположенных в различных географических точках и соединенных выделенными каналами (рис.4.20), которые могут быть как цифровыми, так и аналоговыми.



Для выполнения операций сборки нескольких низкоскоростных потоков байт от алфавитно-цифровых терминалов в пакеты, передаваемые

по сети и направляемые компьютерам для обработки, и обратной разборки пакетов, в сети используются специальные устройства – *PAD (Packet Assembler Disassembler – Сборщик-разборщик пакетов)*, которые в русскоязычных источниках называются ПАД (Пакетный Адаптер Данных). ПАД могут быть встроенными или удалёнными.

Стандартом определён *трехуровневый стек протоколов* с использованием на канальном и сетевом уровнях протоколов *с установлением соединения*, управляющих потоками данных и исправляющих ошибки.

Сетевой уровень рассчитан на работу только *с одним протоколом канального уровня* и не может подобно протоколу IP объединять разнородные сети.

Функциями ПАД в соответствии со стандартом X.3 являются:

- сборка символов, полученных от асинхронных терминалов T, в пакеты;
- разборка пакетов и вывод данных на асинхронные терминалы;
- управление процедурами установления соединения и разъединения по сети X.25;
- передача символов по требованию асинхронного терминала и др.

Терминалы не имеют конечных адресов сети X.25. Адрес присваивается порту ПАД, который подключен к коммутатору пакетов X.25 с помощью выделенного канала.

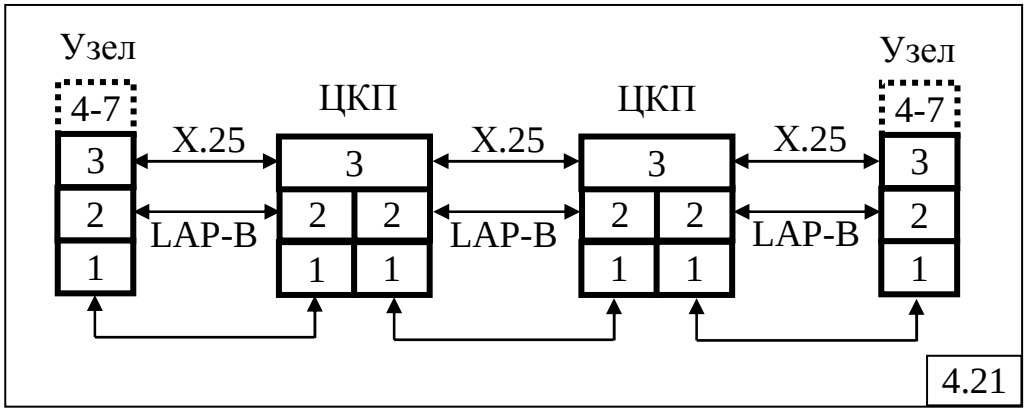
Компьютеры и локальные сети подключаются к сети X.25 непосредственно через адаптер X.25 или маршрутизатор (рис.4.20) с поддержкой протоколов X.25.

4.3.2.2. Стек протоколов сети X.25

Стандарты сетей X.25 описывают 3 уровня протоколов:

- физический;
- канальный;
- сетевой.

На рис.4.21 показана модель взаимодействия конечных узлов (ЭВМ, маршрутизаторы, ПАД) и центров коммутации пакетов (ЦКП).



На *физическом уровне* определён протокол X.21 – универсальный интерфейс между оконечным оборудованием (DTE) и аппаратурой

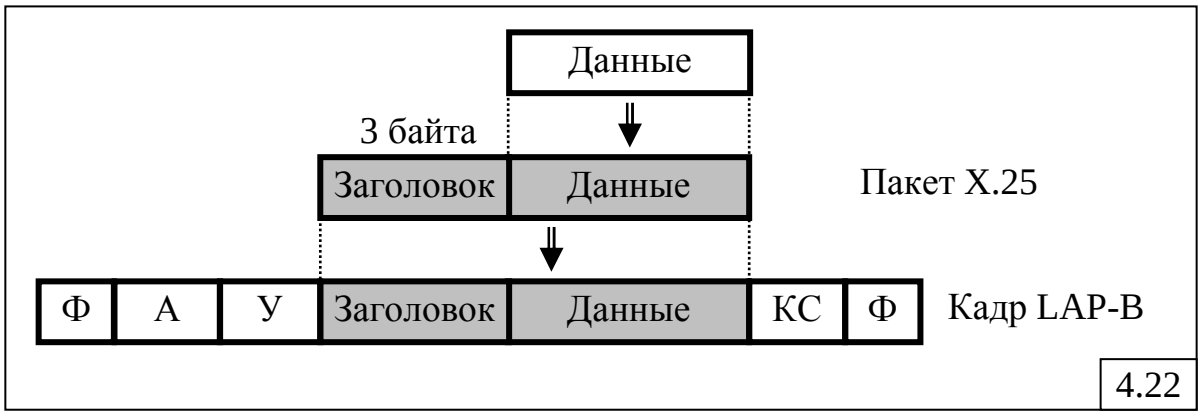
передачи данных (DCE) для синхронного режима работы в сетях общего пользования, а также протокол X.21bis для модемов, удовлетворяющих рекомендациям серии V.

На канальном уровне используется протокол LAP-B, являющийся подмножеством протокола HDLC. Этот протокол обеспечивает сбалансированный режим работы, что означает равноправие узлов, участвующих в соединении. По протоколу LAP-B устанавливается соединение между конечными узлами (компьютером, маршрутизатором или сборщиками-разборщиками пакетов) и коммутатором сети, а также между непосредственно связанными коммутаторами.

Сетевой уровень X.25/3 (в стандарте назван *пакетным уровнем*) реализуется с использованием различных типов пакетов и выполняет функции *маршрутизации пакетов*, *установления* и *разрыва виртуального канала* между конечными абонентами сети и управления потоком пакетов.

На рис.4.22 показана последовательность формирования кадра канального уровня LAP-B, передаваемого между узлами и ЦКП.

В конечных узлах данные более высоких уровней (4-7) упаковываются на сетевом (пакетном) уровне в пакет X.25, который затем передаётся на 2-й канальный уровень, где пакет вкладывается в поле данных кадра LAP-B. Кадр LAP-B включает в себя двухбайтовый заголовок, содержащий адрес (А) и поле «Управление (У)», и концевик (2 или 4 байта), содержащий контрольную сумму (КС). В качестве обрамления кадра используется 8-битовая последовательность 01111110, называемая флагом (Ф). Назначение и содержание указанных полей рассматривается в п.4.4.10.2, посвящённом описанию протокола HDLC.



4.3.2.3. Установление виртуального соединения

Для установления виртуального соединения узел-отправитель посылает узлу-получателю пакет Call Request (рис.4.23) протокола X.25.

Поля, расположенные в первых трех байтах заголовка пакета, используются во всех типах кадров протокола X.25.

Признак **Q** определяет *тип информации* в поле данных пакета: Q=1 – управляющая информация, Q=0 – данные.

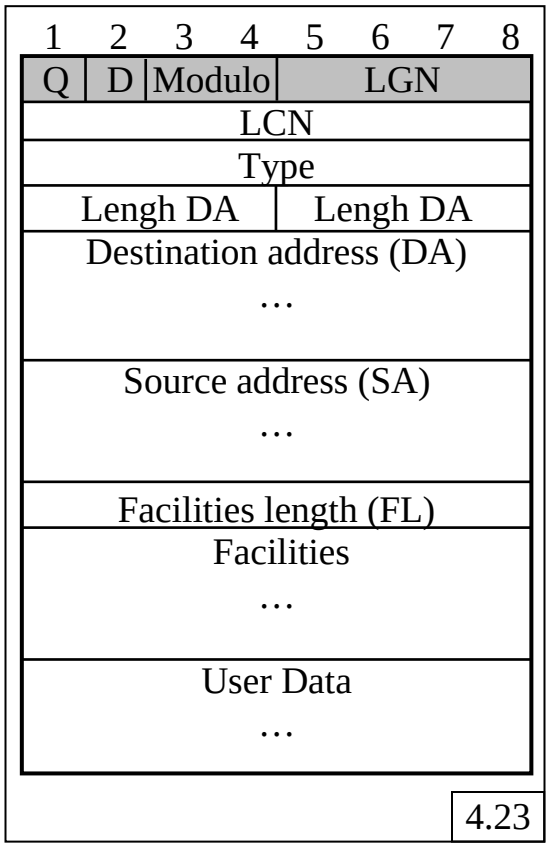
Признак **D** предназначен для *подтверждения приёма* пакета узлом назначения.

Двухбитовое поле **Modulo** задаёт модуль нумерации пакетов: 10 – модуль 128, а 01 – модуль 8.

Поле **LGN** (Logical Group Number) содержит значение *номера логической группы*, объединяющей виртуальные каналы с одним общим функциональным признаком, например:

- постоянный виртуальный канал;
- коммутируемый дуплексный виртуальный канал и т.д.

Поле **LCN** (Logical Channel Number) содержит *номер виртуального канала*, назначаемый узлом-источником (для коммутируемых виртуальных каналов) или администратором сети (для постоянных виртуальных каналов). Максимальное количество виртуальных каналов, проходящих через один порт, равно 2^8=256.



4.23

Поле **Тип** (Type), длиной 8 бит, указывает *тип пакета*: управляющий пакет или пакет данных. Для управляющих пакетов в этом же поле указывается подтип пакета, а для пакетов данных – номера положительных и отрицательных квитанций.

Следующие два поля определяют *длину адресов назначения и источника* (**DA** и **SA**), которые располагаются в следующих двух полях. Адрес **DA** используется для маршрутизации пакета Call Request, а **SA** – для передачи узлом назначения подтверждения об установлении соединения путём посылки пакета Call Accepted – «Запрос принят», в котором эти адреса меняются местами. Адреса могут иметь произвольный формат.

Поля **Facilities length** (*Длина поля услуг*) и **Facilities** (*Услуги*) нужны для согласования дополнительных услуг, которые предоставляет сеть абоненту. Например, пользователь с помощью услуги «Согласование параметров управления потоком» может использовать нестандартные значения параметров протокола, таких как размер окна, максимальный размер поля данных пакета и т. п.

Поле **User Data** (*Поле данных*) может иметь различные максимальные значения длины: от 64 до 4096 байт. Предпочтительной является длина 128 байт.

Пакет Call Request маршрутизируется в узлах сети на основании таблицы маршрутизации, прокладывая при этом виртуальный канал. Начальное значение номера виртуального канала задает пользователь в этом пакете в поле LCN. После установления виртуального канала конечные узлы обмениваются пакетами данных (Data), в которых первые

три байта такие же, как в пакете Call Request, а адресные поля и поля услуг отсутствуют.

Коммутаторы (ЦКП) сетей X.25 проще и дешевле маршрутизаторов, поскольку не поддерживают процедур обмена маршрутной информацией и, как следствие, процедур поиска оптимальных маршрутов, а также не выполняют преобразований форматов кадров канальных протоколов. С другой стороны, по сравнению с коммутаторами локальных сетей, которые просто передают поступивший кадр на выходной порт, коммутаторы X.25 выполняют ряд дополнительных функций, а именно:

- принимают кадр LAP-B и проверяют контрольную сумму;
- при обнаружении ошибки или утере кадра организуют повторную передачу;
- формируют ответ-подтверждение с конкретным номером;
- определяют по номеру виртуального канала выходной порт, извлекают из кадра пакет X.25, а затем формируют новый кадр для дальнейшего продвижения пакета.

Наличие этих функций обусловливает сравнительно невысокую производительность коммутаторов X.25, которая составляет несколько тысяч пакетов в секунду.

Протоколы сетей X.25 были разработаны *для низкоскоростных каналов связи с высоким уровнем помех* и *не гарантируют требуемой пропускной способности*, но могут устанавливать приоритет трафика отдельных виртуальных каналов, который указывается в запросе на установление соединения в поле услуг.

4.3.3. Сети Frame Relay

4.3.3.1. Особенности технологии Frame Relay

Frame Relay – сети, которые по сравнению с сетями X.25 гораздо лучше подходят для передачи пульсирующего трафика локальных сетей в тех случаях, когда каналы связи приближаются по качеству к каналам локальных сетей, например при использовании волоконно-оптических кабелей.

Рассмотрим кратко основные особенности технологии Frame Relay.

1. *Низкая протокольная избыточности* и *дейтаграммный режим работы* сетей Frame Relay обеспечивает высокую пропускную способность (до 2 Мбит/с) и небольшие задержки кадров. В то же время технология Frame Relay *не обеспечивает надежную передачу* кадров, возлагая эти функции на протоколы верхних уровней.

2. *Гарантированная поддержка основных показателей качества обслуживания* – средней скорости передачи данных по виртуальному каналу при допустимых пульсациях трафика – основная особенность, отличающая технологию Frame Relay от X.25.

3. Стандарты Frame Relay определяют *два типа виртуальных каналов* – постоянные (PVC) и коммутируемые (SVC).

4. Технология Frame Relay использует для передачи данных технику виртуальных соединений, аналогичную той, которая применяется в сетях X.25. Однако пользовательские данные (при установленном виртуальном соединении) в сетях Frame Relay передаются по протоколам только *физического и канального уровней*, в то время как в сетях X.25 после установления соединения данные передаются протоколом 3-го уровня.

5. По сравнению с технологией X.25 в сетях Frame Relay меньше *накладные расходы* при передаче данных, так как они вкладываются в кадры канального уровня, а не в пакеты сетевого уровня, как в сетях X.25.

6. Протокол канального уровня LAP-F в сетях Frame Relay, относящийся к семейству протоколов HDLC, имеет *два режима работы* – **основной (core) и управляющий (control)**. В основном режиме кадры передаются без преобразования и контроля, как и в коммутаторах локальных сетей. За счет этого сети Frame Relay обладают весьма высокой производительностью, так как кадры в коммутаторах не подвергаются преобразованию, а сеть не передает квитанции подтверждения между коммутаторами на каждый пользовательский кадр, как это происходит в сети X.25. Пульсирующий трафик передаётся в сети Frame Relay достаточно быстро и без больших задержек.

7. Технология Frame Relay, ориентированная на использование каналов связи высокого качества, не предусматривает выполнение функций по *обнаружению и коррекции искажённых кадров*. Эти функции возлагаются на конечные узлы, которые должны обнаруживать и корректировать ошибки с использованием протоколов транспортного или более высоких уровней. В этом отношении технология Frame Relay близка к технологиям локальных сетей, таким как Ethernet, Token Ring и FDDI, которые тоже только *отбрасывают искаженные кадры*, но сами не занимаются их повторной передачей.

Способность технологии Frame Relay *гарантировать некоторые параметры качества обслуживания* (QoS) является ключевой. Именно поэтому данная технология получила широкое распространение.

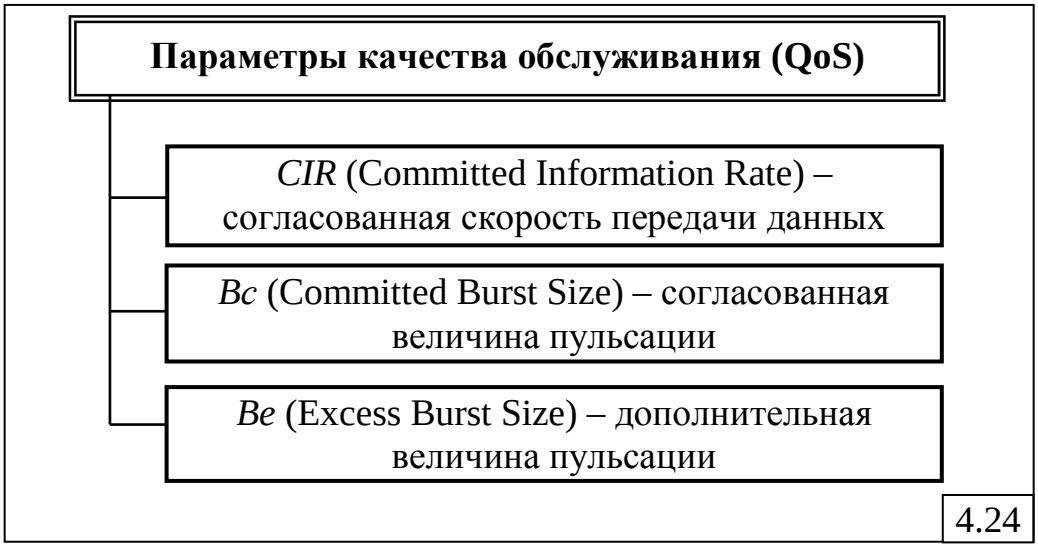
4.3.3.2. Поддержка качества обслуживания

Технология Frame Relay гарантированно обеспечивает выполнение основных параметров качества транспортного обслуживания, необходимых при объединении локальных сетей. Для этого при установлении соединения используется **процедура заказа качества обслуживания**, отсутствующая в сетях X.25 и заключающаяся в следующем.

Для каждого виртуального соединения определяются значения параметров, влияющих на качество обслуживания (рис.4.24):

- *CIR* (Committed Information Rate) – ***согласованная информационная скорость***, с которой сеть будет передавать данные пользователя;

- *Bc* (Committed Burst Size) – ***согласованный объем пульсации***, то есть максимальное количество байтов, которое сеть будет передавать от этого пользователя за интервал времени *T*;
- *Be* (Excess Burst Size) – ***дополнительный объем пульсации***, то есть максимальное количество байтов, которое сеть будет пытаться передать сверх установленного значения *Bc* за интервал времени T.



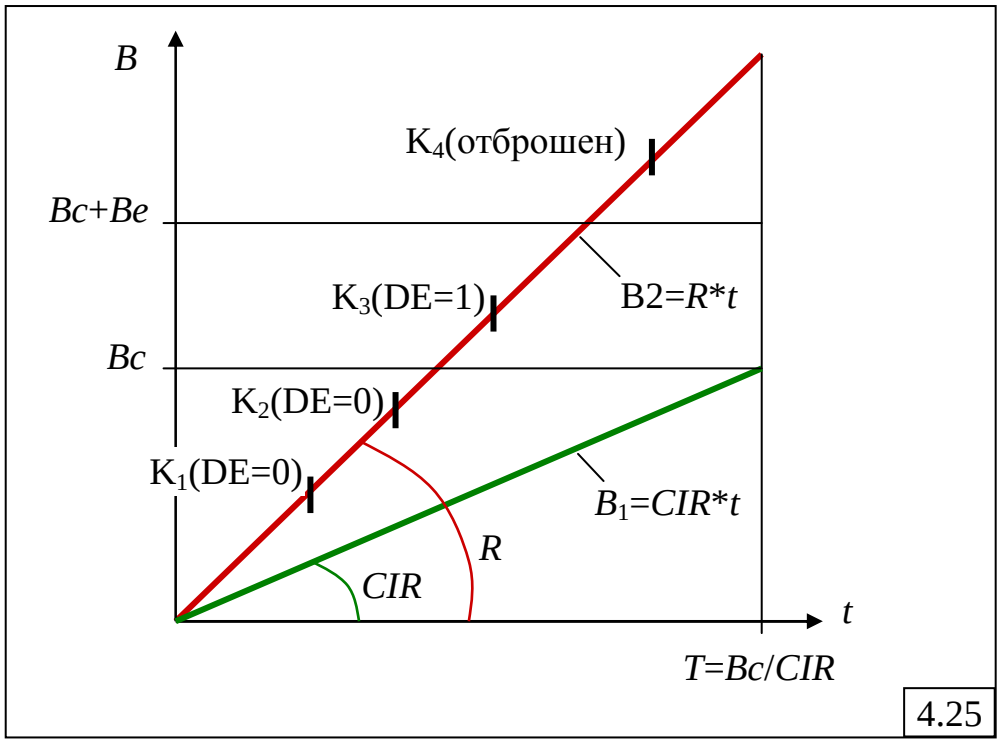
Гарантий по *задержкам* передачи кадров технология Frame Relay не дает, оставляя эту услугу сетям ATM.

Основным параметром, по которому абонент и сеть заключают соглашение при установлении виртуального соединения, является *согласованная скорость передачи данных*. Для постоянных виртуальных каналов это соглашение является частью контракта на пользование услугами сети. При установлении коммутируемого виртуального канала соглашение о качестве обслуживания заключается автоматически с помощью протокола Q.931/933 – требуемые параметры *CIR*, *Bc* и *Be* передаются в пакете запроса на установление соединения.

Так как скорость передачи данных можно измерить только на каком-то интервале времени, то в качестве такого контрольного интервала, на котором проверяются условия соглашения, выбирается время T, значение которого определяется следующим образом: $T = Bc / CIR$ (рис.4.25).

Пользователь в соответствии с соглашением должен передавать в сеть данные со средней скоростью, равной *CIR* (прямая $B_1 = CIR * t$ на рис.4.25). Если же он нарушает соглашение и передаёт данные со средней скоростью *R* (прямая $B_2 = R * t$ на рис.4.25), то сеть не гарантирует доставку кадра. При этом, до тех пор, пока объём переданных данных не превышает *Bc* кадры имеют специальный признак DE (Discard Eligibility), равный 0 (кадры $К_1$ и $К_2$). Если же объём переданных данных превысил *Bc*, то все последующие кадры помечаются признаком DE, равным 1 (кадр $К_3$). Кадры, отмеченные таким признаком, подлежат удалению, однако они удаляются из сети только в том случае, если коммутаторы будут

перегружены. Если же перегрузок нет, то кадры с признаком DE=1 доставляются адресату.



Такое поведение сети соответствует случаю, когда общей объём данных, переданных пользователем в сеть за период T, не превышает (*Bc*+*Be*). Если же этот порог превышен, то кадр не помечается признаком DE, а немедленно удаляется из сети (кадр K_4).

Для контроля соглашения о параметрах качества обслуживания все коммутаторы сети Frame Relay выполняют так называемый алгоритм «дырявого ведра» (Leaky Bucket). Алгоритм использует *счетчик* поступивших от пользователя байт. Каждые *T* секунд значение счетчика уменьшается на величину *Bc* или же сбрасывается в 0, если значение счетчика меньше, чем *Bc*. Все кадры, данные которых не увеличили значение счетчика свыше порога *Bc*, пропускаются в сеть со значением признака DE=0. Кадры, которые увеличили значение счетчика свыше *Bc*, но меньше (*Bc*+*Be*), также передаются в сеть, но с признаком DE=1. И наконец, кадры, которые увеличили значение счетчика свыше (*Bc*+*Be*), отбрасываются коммутатором.

Пользователь может включить в соглашение не все параметры качества обслуживания, а только некоторые. Например, использование параметров *CIR* и *Bc* обеспечивает более качественное обслуживание, так как кадры никогда не отбрасываются коммутатором сразу. Коммутатор только помечает признаком DE=1 кадры, которые превышают порог *Bc* за время *T*. Если в сети не возникают перегрузки, то кадры такого канала всегда дойдут до конечного узла, даже если пользователь нарушает соглашение с сетью.

Механизм заказа средней пропускной способности и максимальной пульсации является основным механизмом управления потоками кадров в

сетях Frame Relay. Соглашения должны заключаться таким образом, чтобы сумма средних скоростей виртуальных каналов не превосходила возможностей портов коммутаторов. При заказе постоянных каналов за это отвечает администратор, а при установлении коммутируемых виртуальных каналов – программное обеспечение коммутаторов. При правильно взятых на себя обязательствах сеть борется с перегрузками путем удаления кадров с признаком DE=1 и кадров, превысивших порог (*Bc*+*Be*).

Кроме этого, в технологии Frame Relay определен ещё и дополнительный (необязательный) механизм управления кадрами. Это механизм оповещения конечных пользователей о перегрузках в коммутаторах сети.

При создании коммутируемого виртуального канала параметры качества обслуживания передаются в сеть с помощью протокола Q.931. Этот протокол устанавливает виртуальное соединение с помощью нескольких служебных пакетов.

4.3.3.3. Использование сетей Frame Relay

Услуги Frame Relay и X.25 обычно предоставляются одними и теми же операторами, а производители выпускают коммутаторы, которые могут работать как по протоколам X.25, так и по протоколам Frame Relay.

Технология Frame Relay в территориальных сетях с коммутацией пакетов можно рассматривать как аналог технологии Ethernet в локальных сетях. Обе технологии:

- предоставляют быстрые базовые транспортные услуги, доставляя кадры без гарантий в узел назначения дейтаграммным способом;
- если кадры теряются, то не предпринимаются никакие усилия для их восстановления.

Отсюда вывод – *полезная пропускная способность при работе через сети Frame Relay зависит от качества каналов и методов восстановления пакетов* на уровнях стека протоколов, расположенного над протоколом Frame Relay. Если каналы качественные, то кадры будут теряться и искажаться редко, так что скорость восстановления пакетов протоколами транспортного уровня будет вполне приемлема. Если же кадры искажаются и теряются часто, то полезная пропускная способность в сети Frame Relay может упасть в десятки раз, как это происходит в сетях Ethernet при плохом состоянии кабельной системы. Поэтому сети Frame Relay следует применять при наличии на магистральных каналах волоконно-оптических кабелей высокого качества. Каналы доступа могут быть и на витой паре, при условии обеспечения приемлемого уровня искажения данных.

Отсутствие гарантий на задержку передачи кадров в сетях Frame Relay и сравнительно небольшая скорость передачи данных в 2 Мбит/с ограничивают их применение для передачи голоса и практически делают невозможным передачу видео.

Для передачи голоса в сетях Frame Relay используется приоритезация трафика, заключающаяся в присвоении кадрам, переносящим замеры голоса, *приоритетов*. Магистральные коммутаторы Frame Relay такие кадры обслуживают в первую очередь.

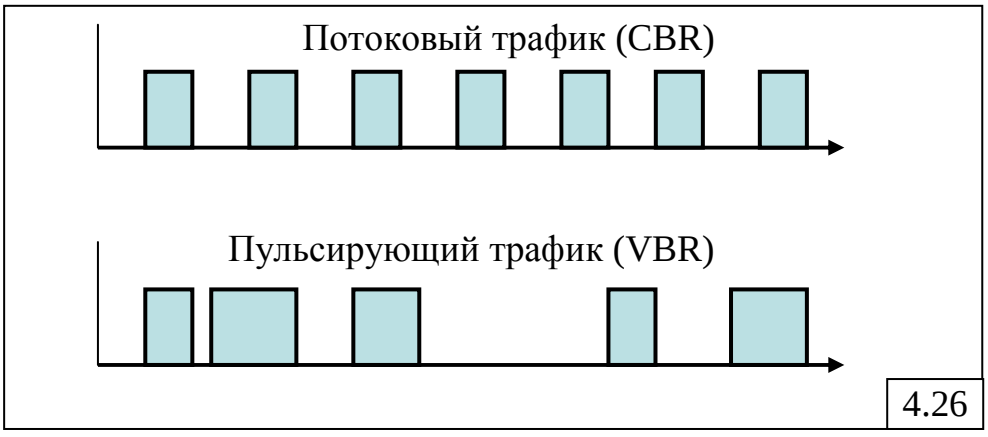
4.3.4. Технология ATM

ATM (Asynchronous Transfer Mode) - технология *асинхронного режима передачи*, использующая маленькие пакеты фиксированного размера, называемые *ячейками* (cells), предназначенная для передачи в сети *различных видов трафика* – голос, видео и данные, обеспечивая при этом достаточную пропускную способность для каждого из них и гарантируя своевременную доставку восприимчивых к задержкам данных. Технология ATM может использоваться как для построения высокоскоростных локальных сетей, так и магистралей, объединяющих традиционные локальные сети.

ATM разрабатывалась как альтернатива *синхронной передаче* STM (Synchronous Transfer Mode), в основе которой лежит технология TDM. Главный недостаток технологии TDM заключается в невозможности перераспределять пропускную способность объединенного канала между подканалами (временными слотами), которые предоставляются пользователям для передачи данных. Если временной слот не используется пользователем и подканал свободен, его ресурсы не могут быть переданы другому пользователю, что приводит к потере пропускной способности канала и, как следствие, к снижению реальной скорости передачи данных. В технологии ATM ячейки не привязаны к временны*м* слотам, а их идентификация на приёмной стороне осуществляется не по номеру слота, а по идентификатору виртуального соединения.

Трафик современных компьютерных сетей можно разбить на два больших класса:

- потоковый (stream), представляющий собой равномерный поток данных (рис.4.26,а) с постоянной битовой скоростью (CBR – Constant Bit Rate);
- пульсирующий (burst), представляющий собой неравномерный непредсказуемый поток данных (рис.4.26,б) с переменной битовой скоростью (VBR – Variable Bit Rate).



4.26

Потоковый трафик характерен для аудио и видео данных, для которых основной характеристикой качества обслуживания является задержка передачи данных. Пульсирующий трафик формируется приложениями, связанными, например, с передачей файлов и при работе пользователей в режиме «запрос-ответ». Пульсирующий трафик обычно нечувствителен к задержкам, но чувствителен к потерям и искажениям передаваемых пакетов.

Технология ATM разрабатывалась как технология, способная обслуживать все виды трафика в соответствии с их требованиями за счёт использования:

- техники виртуальных каналов;
- предварительного заказа параметров качества обслуживания;
- приоритезации трафика.

Стандарты определяют ATM как *интерфейс и протокол*, которые разработаны *для коммутации* трафика через общую *высокоскоростную* среду с постоянной или переменной битовой скоростью.

4.3.4.1. Общие принципы технологии ATM

Подход, реализованный в технологии ATM, состоит в передаче любого вида трафика – компьютерного или мультимедийного – пакетами фиксированной длины в 53 байта, называемыми ячейками (cell). Поле данных ячейки занимает 48 байт, а заголовок – 5 байт.

Размер ячеек выбирался исходя из двух противоречивых условий:

- с одной стороны, размер ячейки должен быть достаточно мал, чтобы сократить время задержки в узлах сети;
- с другой стороны, размер ячейки должен быть достаточно велик, чтобы минимизировать потери пропускной способности, обусловленные накладными расходами на передачу заголовка ячейки.

Преимущества ячеек перед кадрами локальных сетей подробно рассмотрены в п.1.5.1.4.

Для уменьшения доли служебной информации в ячейке в технологии ATM применен стандартный для территориально-распределенных вычислительных сетей прием – передача ячеек в соответствии с *техникой виртуальных каналов* с длиной номера виртуального соединения в 24 бит, что вполне достаточно для обслуживания большого количества виртуальных соединений каждым портом коммутатора сети ATM.

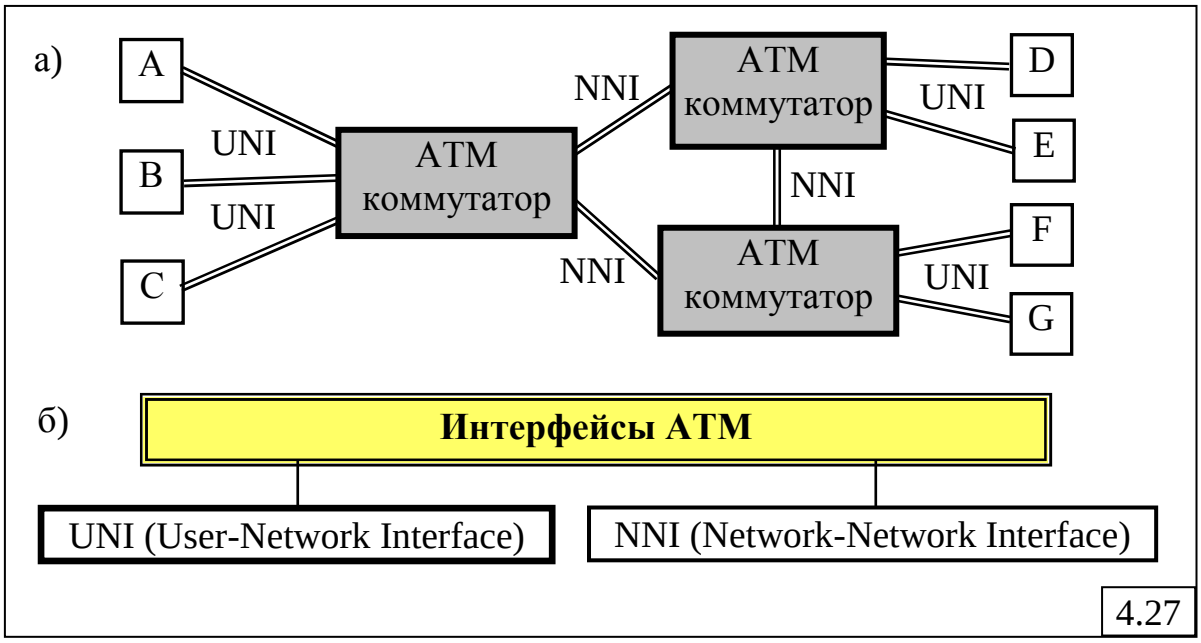
Сеть ATM имеет классическую структуру территориальной сети (рис.4.27,а) – конечные станции A, B, ..., G соединяются индивидуальными каналами с коммутаторами, которые в свою очередь могут соединяться с другими коммутаторами. Соответственно в стандарте определены 2 типа интерфейса (рис.4.27,б):

- пользователь – сеть (User-Network Interface, UNI);
- сеть – сеть (Network-Network Interface, NNI).

Спецификация UNI определяет:

- структуру пакета,

- адресацию станций,
- обмен управляющей информацией,
- уровни протокола ATM,
- способы установления виртуального канала,
- способы управления трафиком.



Коммутация пакетов происходит на основе ***идентификатора виртуального канала*** (Virtual Channel Identifier, VCI), который назначается соединению при его установлении и уничтожается при разрыве соединения.

Виртуальные каналы могут быть *постоянными* (PVC) и *коммутируемыми* (SVC). Для ускорения коммутации в больших сетях используется понятие *виртуального пути (Virtual Path)*, который объединяет виртуальные каналы, имеющие в сети ATM общий маршрут между исходным и конечным узлами или общую часть маршрута между двумя коммутаторами сети. ***Идентификатор виртуального пути*** (Virtual Path Identifier, VPI) является старшей частью локального адреса и представляет собой общий префикс для некоторого количества различных виртуальных каналов. Таким образом, адресация в технологии ATM реализована на двух уровнях:

- на уровне адресов конечных узлов (на этапе установления виртуального канала);
- на уровне номеров виртуальных каналов (при передаче данных по сформированному виртуальному каналу).

Стандарт ATM не вводит свои спецификации на реализацию физического уровня и основывается на технологии SDH/SONET, принимая её иерархию скоростей. Организация ATM Forum определила для ATM не все иерархии скоростей SDH, а только скорости OC-3 (155 Мбит/с) с использованием волоконно-оптического кабеля или неэкранированной

витой пары категории 5 и OC-12 (622 Мбит/с) с использованием только волоконно-оптического кабеля.

Имеются и другие физические интерфейсы сетей ATM, отличные от SDH/SONET:

- интерфейсы T1/E1 и T3/E3, используемые в глобальных сетях;
- интерфейсы локальных сетей со скоростью 100 Мбит/с (FDDI) и 25 Мбит/с.

Для решения задачи совмещения разнородного трафика в одной сети в технологии ATM реализован принцип *заказа пропускной способности* и *качества обслуживания*, как в технологии Frame Relay.

4.3.4.2. Стек протоколов ATM

Стек протоколов ATM показан на рис.4.28, а распределение протоколов по конечным узлам и коммутаторам ATM – на рис.4.29.

<table>
<tr><td colspan="3">Верхние уровни</td></tr>
<tr><td rowspan="3">Уровни адаптации ATM (AAL1-5)</td><td rowspan="2">Подуровень конвергенции</td><td>Общая часть подуровня конвергенции</td></tr>
<tr><td>Специфическая для сервиса часть</td></tr>
<tr><td colspan="2">Подуровень сегментации и реассемблирования</td></tr>
<tr><td>Уровень ATM</td><td colspan="2">(маршрутизация, мультиплексирование, управление потоком, обработка приоритетов)</td></tr>
<tr><td rowspan="2">Физический уровень</td><td colspan="2">Подуровень согласования передачи</td></tr>
<tr><td colspan="2">Подуровень, зависящий от физической среды</td></tr>
</table>

4.28

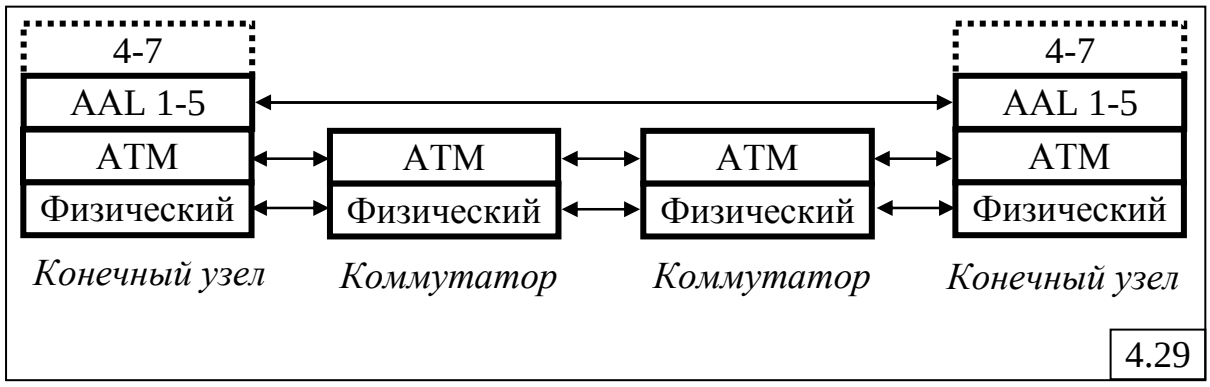
Стек протоколов ATM соответствует нижним уровням семиуровневой модели ISO/OSI и включает:

- уровень адаптации ATM,
- собственно уровень ATM;
- физический уровень.

Прямого соответствия между уровнями протоколов технологии ATM и уровнями модели OSI нет.

Уровень адаптации (ATM Adaptation Layer, AAL) представляет собой набор протоколов, которые преобразуют блоки данных протоколов верхних уровней сети ATM в ячейки ATM нужного формата. Функции этих уровней достаточно условно соответствуют функциям транспортного уровня модели OSI, например функциям протоколов TCP или UDP. Протоколы AAL при передаче пользовательского трафика работают только

в конечных узлах сети (см. рис.4.29), как и транспортные протоколы большинства технологий.



Уровень ATM занимает в стеке протоколов ATM примерно то же место, что протокол IP в стеке TCP/IP или протокол LAP-F в стеке протоколов технологии Frame Relay. Протокол ATM занимается передачей ячеек через коммутаторы при установленном и настроенном виртуальном соединении, то есть на основании готовых таблиц коммутации портов. Протокол ATM выполняет коммутацию по номеру виртуального соединения, который в технологии ATM разбит на две части – *идентификатор виртуального пути (Virtual Path Identifier, VPI)* и *идентификатор виртуального канала (Virtual Channel Identifier, VCI).* Кроме этой основной задачи протокол ATM выполняет ряд функций по контролю за соблюдением трафик-контракта со стороны пользователя сети, маркировке ячеек-нарушителей, отбрасыванию ячеек-нарушителей при перегрузке сети, а также управлению потоком ячеек для повышения производительности сети.

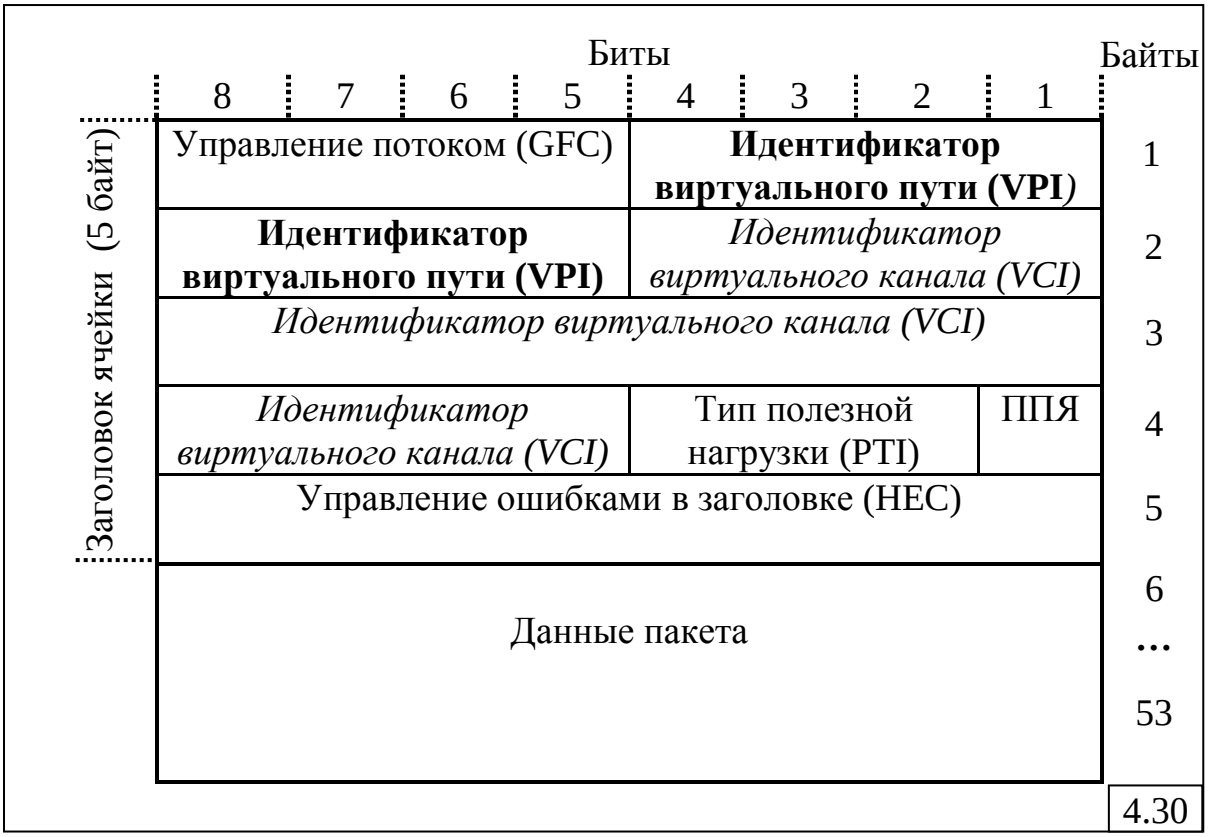
4.3.4.3. Формат ATM-ячейки

Протокол ATM работает с ячейками следующего формата, представленного на рис.4.30.

Поле **Управление потоком** *(Generic Flow Control)* используется только в UNI при взаимодействии конечного узла и первого коммутатора сети для управления трафиком и предотвращения перегрузки. Для NNI это поле не определено, а его биты используются для расширения поля идентификатора виртуального пути (VPI).

Поля **Идентификатор виртуального пути** (VitualPath Identifier, VPI) и **Идентификатор виртуального канала** (Vitual Channel Identifier, VCI) занимают соответственно 8 и 16 бит. Эти поля задают ***номер виртуального соединения***, разделенный на старшую (VPI) и младшую (VCI) части.

Поле **Тип полезной нагрузки** *(Payload Type Identifier, PTI)* состоит из 3-х бит и задает тип полезной нагрузки, переносимой ячейкой – пользовательские данные или управляющая информация (например, для установления виртуального соединения). Кроме того, один бит этого поля используется для указания перегрузки в сети.



В однобитовом поле **ППЯ** – *Приоритет Потери Ячейки (Cell Loss Priority, CLP)* коммутаторы АТМ отмечают ячейки, которые нарушают соглашения о параметрах качества обслуживания, чтобы удалить их при перегрузках сети: ячейки с CLP=0 являются высокоприоритетными, а ячейки с CLP=1 – низкоприоритетными и могут быть удалены при перегрузках.

Поле **Управление ошибками в заголовке** *(Header Error Control, HEC)* содержит контрольную сумму, вычисленную *для заголовка* ячейки.

4.3.4.4. Принцип работы коммутаторов АТМ

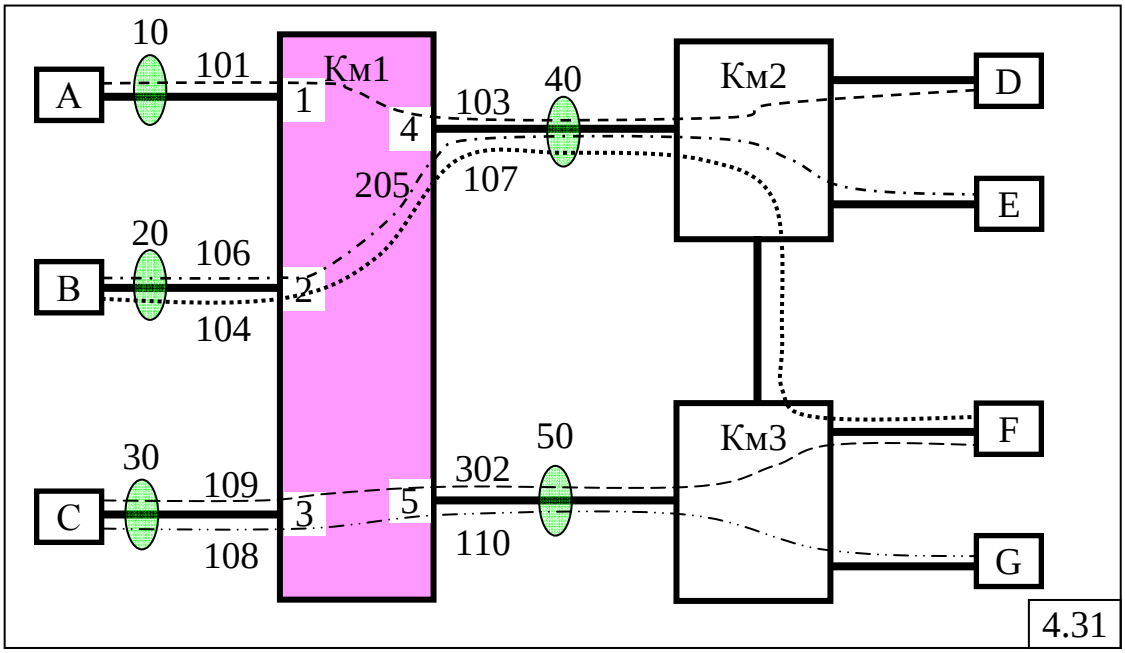
Проиллюстрируем принцип работы коммутаторов АТМ на примере АТМ-сети, представленной на рис.4.27,а). Для простоты положим, что узлы А, В, … представляют собой оконечные коммутаторы, к которым подключены соответствующие пользователи (абоненты) сети. Это означает, что между узлами А, В, … и коммутаторами Км1, Км2 и Км3 данные передаются в виде ячеек.

Положим, что в процессе установления соединения, сформированы виртуальные соединения, показанные на рис.4.31 и построена таблица коммутации для коммутатора Км1, представленная на рис.4.32.

Как видно из таблицы, сформировано 5 виртуальных соединений (каналов) между абонентами сети: A – D, B – E, B – F, C – F и C – G.

Рассмотрим процесс прохождения через Км1 ячейки от абонента А, в заголовке которой в момент её поступления в порт 1 коммутатора в качестве идентификаторов виртуального пути и виртуального канала будут находиться значения VPI=10 и VCI=101 (рис.4.31). В соответствии с

записью в первой строке таблицы коммутации, поступившая на 1-й порт ячейка с VPI=10 и VCI=101 должна быть направлена в 4-й порт коммутатора, причём в заголовке ячейки идентификаторы виртуального пути и виртуального канала должны быть заменены на значения VPI=40 и VCI=103 (рис.4.33). Аналогично, ячейка, поступившая на 2-й порт с VPI=20 и VCI=104 будет направлена в 4-й порт коммутатора, причём в заголовке идентификаторы виртуального пути и виртуального канала будут заменены на значения VPI=40 и VCI=107.

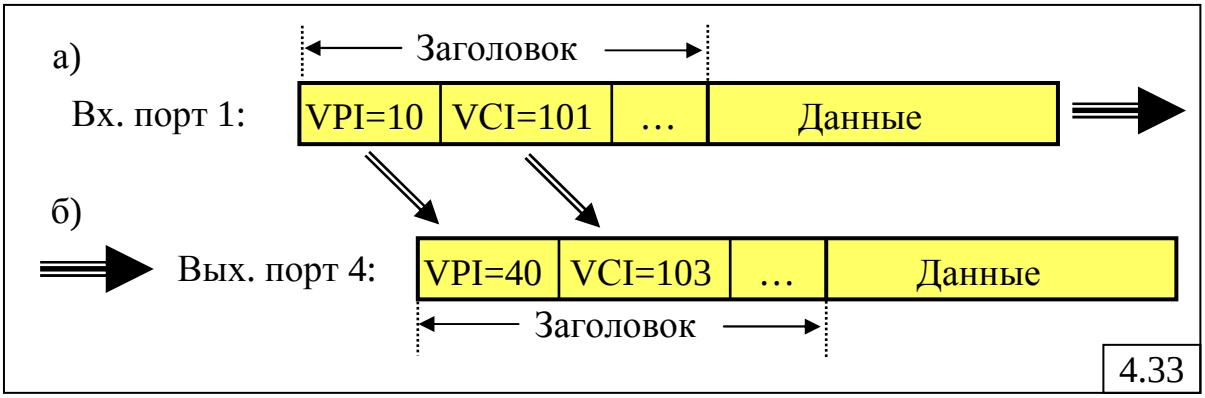


4.31

Таблица коммутации коммутатора Км1

Вход			*Выход*		
Порт	VPI	VCI	Порт	VPI	VCI
1	10	101	4	40	103
4	40	205	2	20	106
2	20	104	4	40	107
3	30	108	5	50	110
3	50	302	3	30	109

4.32



4.33

4.3.4.5. Обеспечение качества обслуживания

Качество обслуживания (QoS) в ATM-сетях задаётся следующими параметрами трафика виртуального соединения:

- пиковая скорость передачи ячеек (Peak Cell Rate, PCR);
- средняя скорость передачи ячеек (Sustained Cell Rate, SCR);
- минимальная скорость передачи ячеек (Minimum Cell Rate, MCR);
- максимальная величина пульсаций (Maximum Burst Size, MBS);
- доля потерянных ячеек (Cell Loss Ratio, CLR);
- задержка ячеек (Cell Transfer Delay, CTD);
- вариация задержек ячеек (Cell Delay Variation, CDV).

В зависимости от требований, предъявляемых к качеству передачи данных, в ATM-сетях различают 5 классов трафика, различающихся:

- скоростью передачи;
- чувствительностью к задержкам;
- способом установления соединения;
- совокупностью параметров QoS, характерных для данного класса.

В табл.4.2 представлена классификация классов трафика в соответствии с указанными признаками и приведены примеры трафика каждого класса. Здесь же представлен тип протокола (AAL1-AAL5) уровня адаптации ATM (AAL), который обеспечивает реализацию заданных требований.

Таблица 4.2

Класс	**A**	**B**	**C**	**D**	**X**
Скорость	Постоянная	Переменная			
К задержке	Чувствительны		Не чувствительны		
Соединение	С установлением			Без установления	
Примеры трафика	Голос, ТВ	Компрессирован. голос, ТВ	Компьютерные данные	Трафик компьютерных сетей	
Параметры QoS	PCR, CTD,CDV	PCR, SCR, MBS, CTD, CDV	PCR,SCR,MBS	Не определены	Определяются пользователем
AAL	AAL1	AAL2	AAL5	AAL3/4	

В представленной классификации предусмотрен дополнительный класс трафика, отличающийся от классов A, B, C и D, параметры которого могут быть определены пользователем.

4.3.4.6. Использование технологии ATM

Основной соперник технологии ATM в локальных сетях – гигабитные технологии Ethernet. Там, где необходима высокоскоростная магистраль и не требуется поддержка QoS разных типов трафика, целесообразно использовать технологию Gigabit Ethernet. Технология ATM может оказаться предпочтительней там, где важно обеспечить заданное качество обслуживания (видеоконференции, трансляция телевизионных передач и т. п.).

В территориально-распределенных сетях ATM применяется там, где сеть Frame Relay не справляется с большими объемами трафика, и там, где нужно обеспечить низкий уровень задержек, необходимый для передачи информации реального времени.

4.4. Глобальная сеть Internet

Глобальная сеть Internet реализована на основе стека сетевых протоколов TCP/IP, обеспечивающих передачу данных между разнородными локальными и территориальными сетями, а также коммуникационными системами и устройствами.

4.4.1. Краткая история создания и организационные структуры Internet

Появлению сети Internet и стека протоколов TCP/IP предшествовала в середине 1960-х годов разработка под эгидой агентства DARPA (Defence Advanced Research Projects Agency – Управление перспективных исследований Министерства обороны США) сети, получившей название ARPANET (Advanced Research Projects Agency NETwork). Разработка сети была поручена Стэндфордскому исследовательскому институту и трём американским университетам: Калифорнийскому в Лос-Анжелесе и Университетам штата Юта и штата Калифорния в Санта-Барбаре. Экспериментальная сеть из четырёх узлов была запущена в конце 1969 года, а к концу 1972 года в сети насчитывалось более 30 узлов.

В 1974 году были разработаны модели и протоколы TCP/IP для управления обменом данными в интерсетях, а 1 января 1983 года сеть ARPANET полностью перешла на протокол TCP/IP.

В конце 1970-х годов Национальный научный фонд США (National Science Foundation, NSF) начал разработку межуниверситетской сети, получившей название NSFNet, которая имела гораздо бóльшую пропускную способность, чем ARPANET. В середине 1980-х годов произошло объединение сетей NSFNet и ARPANET, за которым закрепилось название INTRNET (Интернет).

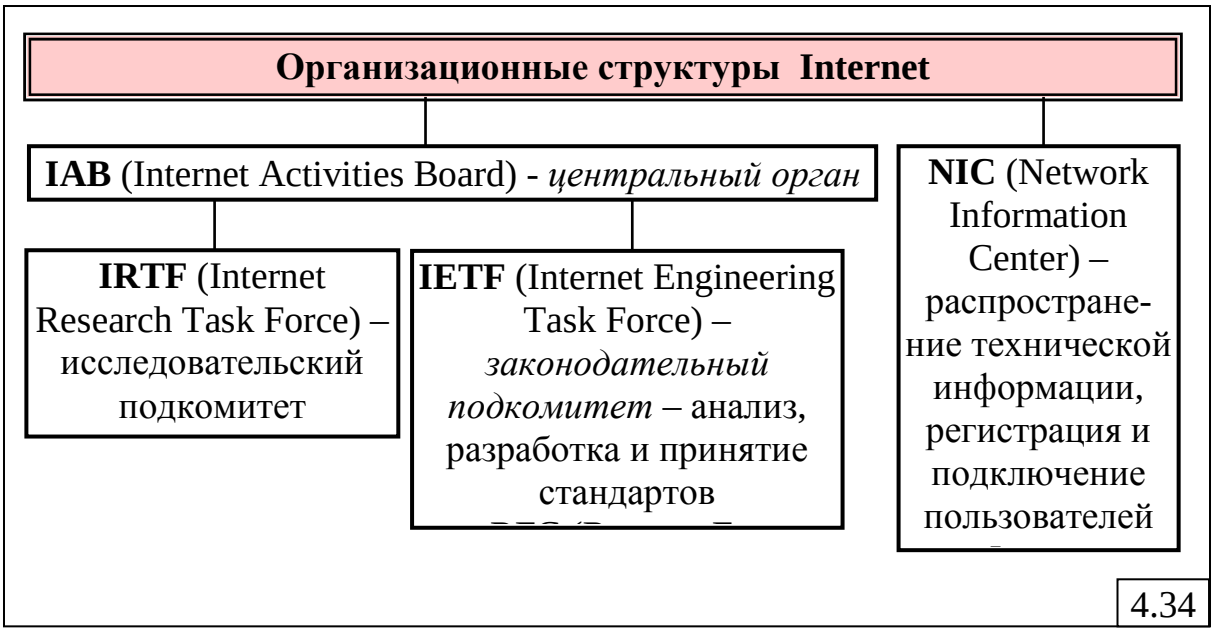
В 1984 году была разработана система доменных имён (Domain Name System, DNS), а в 1989 году появилась концепция Всемирной паутины (World Wide Web,WWW) и были разработаны протокол передачи гипертекста HTTP (HyperText Transfer Protocol) и язык разметки гипертекста HTML (HyperText Markup Language).

Благодаря отсутствию единого руководства и открытости технических стандартов Интернет объединил большинство существующих сетей и к началу 21 века стал популярным средством для обмена данными.

В настоящее время подключиться к Интернету можно через спутники связи, радио-каналы, кабельное телевидение, телефон, сотовую связь, специальные оптико-волоконные линии или электропровода.

Координация разработок и поддержка Интернета осуществляется следующими **организационными структурами** (рис.4.34):

- Internet Activities Board (IAB) – центральный орган, включающий два подкомитета:
  - *исследовательский* – IRTF (Internet Research Task Force);
  - *законодательный* – IETF (Internet Engineering Task Force), выполняющий функцию анализа, разработки и принятия стандартов сети Internet, получивших название RFC (Request For Comments);
- Network Information Center (NIC) – орган, ответственный за распространение технической информации, работу по регистрации и подключению пользователей к Internet и за решение ряда административных задач, таких как распределение адресов в сети.



4.4.2. Стек протоколов TCP/IP

Под **стеком (семейством) протоколов TCP/IP** в широком смысле обычно понимают весь набор реализаций стандартов RFC.

Соответствие уровней TCP/IP уровням OSI-модели и используемые на каждом уровне основные протоколы стека TCP/IP представлены в табл.4.3.

Модель стека протоколов TCP/IP содержит 4 уровня.

На первом уровне (**Network interface – сетевой интерфейс**) находится аппаратно зависимое программное обеспечение, реализующее передачу данных в той или иной среде. Среда передачи данных может

быть реализована различными способами: от простого двухточечного звена до сложной многоузловой коммуникационной структуры сети X.25 или Frame Relay. Стек протоколов TCP/IP поддерживает все стандартные протоколы физического и канального уровней различных сетевых технологий: Ethernet, Token Ring, FDDI, PPP и другие.

Таблица 4.3

Уровни OSI-модели	**Уровни TCP/IP**	**Протокол**	**Блок данных**
5-7	***4. Application (прикладной)***	FTP, TFTP, BGP, HTTP,DHCP, SNMP, DNS, SIP, SMTP, POP3, IMAP,Telnet, PPTP	Сообщение
4	***3. Transport (транспортный)***	TCP, UDP, RTP	Сегмент, Дейтаграмма
3	***2. Internet (межсетевой)***	IPv4, IPv6 , ICMP, IGMP, ARP, RARP, RIP, OSPF	Пакет
1-2	***1. Network interface (сетевой интерфейс)***	SLIP, HDLC, PPP Ethernet, 802.11 Wi-Fi, 802.16 WiMax, Token ring, FDDI, X.25, Frame relay, ATM	Кадр

На втором уровне (**Internet** – межсетевой) реализуется задача маршрутизации с использованием протокола IP. Вторая важная задача протокола IP – сокрытие аппаратно-программных особенностей среды передачи данных и предоставление вышележащим уровням единого унифицированного и аппаратно независимого интерфейса для доставки данных, что обеспечивает многоплатформенное применение приложений, работающих под TCP/IP.

На третьем уровне (**Transport** – транспортный) решаются задачи надежной доставки пакетов и сохранение их порядка и целостности.

На четвёртом уровне (**Application** – прикладной) находятся прикладные задачи, запрашивающие сервис у транспортного уровня.

Основными особенностями стека протоколов TCP/IP являются:

- независимость от среды передачи данных;
- негарантированная доставка пакетов.

Информационные объекты (данные) передаваемые на разных уровнях в сети Интернет получили следующие наименования:

- ***сообщение (message)*** – блок данных, которым оперирует прикладной уровень, предаваемый от приложения к транспортному уровню с соответствующими этому приложению размером и семантикой;
- ***сегмент (segment)*** – блок данных, которым оперирует протокол TCP на транспортном уровне;
- ***дейтаграмма (datagram)*** – блок данных, которым оперирует протокол UDP на транспортном уровне;

- ***пакет (packet)*** – блок данных, называемый также IP-дейтаграммой, которым оперирует протокол IP на межсетевом уровне;
- ***кадр (frame)*** – аппаратно зависимые блоки данных, полученные в результате упаковки IP-дейтаграмм в формат, приемлемый для данной физической среды передачи данных и передаваемый на нижнем уровне TCP/IP-модели, называемом «сетевым интерфейсом».

Рассмотрим кратко перечисленные в табл.4.3 протоколы стека TCP/IP, некоторые из которых (IP, UDP, TCP, HDLC, PPP) более подробно рассматриваются в последующих параграфах.

4.4.2.1. Протоколы прикладного уровня

FTP (File Transfer Protocol – протокол передачи файлов), предназначенный для передачи файлов в сети и доступа к удалённым хостам, реализует следующие функции:

- подключение к серверам FTP;
- просмотр содержимого каталогов;
- загрузка файлов с сервера или на сервер.

FTP функционирует поверх транспортного протокола TCP и использует порт 20/TCP для передачи данных и порт 21/ TCP для передачи команд. В протоколе FTP предусмотрены возможности аутентификации и передачи файла с прерванного места, если передача файла была прервана по какой-то причине.

TFTP (Trivial File Transfer Protocol – простой протокол передачи файлов) предназначен главным образом для первоначальной загрузки бездисковых рабочих станций. TFTP использует транспортный протокол UDP и порт 69/UDP. В отличие от FTP, протокол TFTP не содержит возможностей аутентификации, хотя возможна фильтрация по IP-адресу.

BGP (Border Gateway Protocol – протокол граничного шлюза) – основной протокол динамической маршрутизации в Интернете, предназначенный для обмена информацией о маршрутах между автономными системами. Функционирует поверх протокола транспортного уровня TCP и использует порт 179/TCP.

HTTP (HyperText Transfer Protocol – протокол передачи гипертекста) предназначен для передачи данных (изначально – в виде гипертекстовых документов) на основе клиент-серверной технологии. HTTP в настоящее время используется во Всемирной паутине для получения информации с веб-сайтов.

DHCP (Dynamic Host Configuration Protocol – протокол динамической конфигурации узла) предназначен для автоматического распределения между компьютерами IP-адресов и конфигурационных параметров, необходимых для работы в сети TCP/IP. Протокол реализуется в так называемом DHCP-сервере по клиент-серверной технологии путём выдачи IP-адреса и конфигурационных параметров в ответ на поступивший запрос от компьютера. Протокол DHCP использует транспортный протокол UDP и порты 67/UDP и 68/UDP.

SNMP (Simple Network Management Protocol – протокол простого управления сетями) предназначен для управления и контроля за сетевыми устройствами и приложениями в сети передачи данных путём обмена управляющей информацией. Протокол SNMP встроен во все сетевые ОС и использует транспортный протокол UDP и порты 161/UDP и 162/UDP.

DNS (Domain Name System – система доменных имён) представляет собой компьютерную распределённую иерархическую систему для получения информации о доменах, чаще всего для получения IP-адреса по символьному имени хоста (компьютера или устройства). Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по одноимённому протоколу. Протокол DNS встроен во все сетевые ОС и использует транспортные протоколы TCP и UDP и, соответственно, порты 53/TCP и 53/UDP.

SIP (Session Initiation Protocol) – протокол установления сеанса, предназначенный для установления и завершения пользовательского интернет-сеанса, включающего обмен мультимедийным содержимым (видео- и аудиоконференции, онлайн-игры).

SMTP (Simple Mail Transfer Protocol) – простой протокол передачи почты, предназначенный для передачи электронной почты в сетях TCP/IP.

POP3 (Post Office Protocol Version 3) – протокол почтового отделения, версия 3, обычно используемый почтовым клиентом в паре с протоколом SMTP для получения сообщений электронной почты с сервера. Протокол POP3 использует транспортный протокол TCP и порт 110/TCP. Альтернативным протоколом для сбора сообщений с почтового сервера является протокол IMAP.

IMAP (Internet Message Access Protocol) – протокол доступа к электронной почте Интернета, как и POP3, служит для работы со входящими письмами, однако обеспечивает ряд дополнительных функций, предоставляя пользователю доступ к хранилищу электронных писем на сервере так, как будто эти письма находятся на его компьютере. POP3 использует транспортный протокол TCP и порт 143/TCP. Для отправки писем используется протокол SMTP.

TELNET (TELetype NETwork) – виртуальный текстовый терминал, предназначенный для реализации текстового интерфейса в сети с использованием транспортного протокола TCP (стандартный порт 23/TCP).

PPTP (Point-to-point tunneling protocol) – туннельный протокол типа точка-точка, позволяющий компьютеру устанавливать защищённое соединение с сервером за счёт создания специального туннеля в незащищённой сети. PPTP инкапсулирует кадры PPP в IP-пакеты для передачи через Интернет и может использоваться для организации туннеля между локальными сетями. PPTP использует TCP-соединение для обслуживания туннеля.

4.4.2.2. Протоколы транспортного уровня

TCP (Transmission Control Protocol) – протокол управления передачей данных с установлением соединения, реализующий обмен данными между двумя узлами на основе некоторого соглашения об управлении потоком данных.

UDP (User Datagram Protocol) – дейтаграммный протокол передачи данных в виде независимых единиц – дейтаграмм (datagram).

RTP (Real-time Transport Protocol) предназначен для передачи трафика реального времени. Заголовок RTP-пакета содержит данные, необходимые для восстановления голоса или видеоизображения в приёмном узле, о типе кодирования информации (JPEG, MPEG и т. п.) а также временную метку и номер пакета. Эти параметры позволяют при минимальных задержках определить порядок и момент декодирования каждого пакета, а также интерполировать потерянные пакеты. В качестве нижележащего протокола транспортного уровня, как правило, используется протокол UDP.

4.4.2.3. Протоколы межсетевого уровня

IP (Internet Protocol) - основной протокол стека TCP/IP, реализующий передачу пакетов по IP-сети от узла к узлу. Протокол IP:

а) не гарантирует:

- доставку пакетов,
- целостность пакетов,
- сохранение порядка потока пакетов;

б) не различает логические объекты (процессы), порождающие поток данных.

Эти задачи решают протоколы *транспортного уровня* TCP и UDP, реализующие различные режимы доставки данных. В отличие от IP протоколы транспортного уровня различают приложения и передают данные от приложения к приложению.

В настоящее время на смену протоколу IP версии 4 (IPv4) приходит протокол версии 6 (IPv6).

ICMP (Internet Control Message Protocol) – межсетевой протокол управляющих сообщений, используемый в основном для передачи сообщений об ошибках и исключительных ситуациях, возникших при передаче данных, а также выполняющий некоторые сервисные функции.

ICMP-сообщения генерируются при нахождении ошибок в заголовке IP пакета, при отсутствии маршрута к адресату, а также используются маршрутизаторами для обновления записей в таблице маршрутизации отправителя и для управления скоростью отправки сообщений отправителем. ICMP-сообщения инкапсулируются в IP пакеты.

ICMP является неотъемлемой частью IP, но при этом не делает протокол IP средством надёжной доставки сообщений. Для этих целей существует протокол TCP.

IGMP (Internet Group Management Protocol) – протокол управления группами Интернета, предназначенный для управления групповой (multicast) передачей данных в IP-сетях версии 4 (IPv4). IGMP используется маршрутизаторами и IP-узлами для организации групп сетевых устройств, а также для поддержки потокового видео и онлайн-игр, обеспечивая эффективное использование сетевых ресурсов.

ARP (Address Resolution Protocol – Протокол разрешения адресов) предназначен для определения физического адреса устройства (MAC-адреса) по его IP-адресу.

RARP (Reverse Address Resolution Protocol – Протокол обратного определения адреса) предназначен для определения IP-адреса устройства по его физическому адресу (MAC-адресу).

RIP (Routing Information Protocol) – протокол маршрутизации типа DVA, реализующий алгоритм обмена информацией о доступных сетях и расстояниях до них путём *периодической* рассылки широковещательных пакетов.

OSPF (Open Shortest Path First) – протокол маршрутизации типа LSA, реализующий алгоритм обмена информацией о состоянии каналов, путём периодического тестирования состояния каналов с соседними маршрутизаторами. Протокол OSPF разработанный для применения в сети Интернет и используется в других больших сетях (NetWare, SNA, XNS, DECNet).

4.4.2.4. Протоколы канального уровня («сетевой интерфейс»)

SLIP (Serial Line IP) – первый стандарт канального уровня для выделенных линий, разработанный специально для стека протоколов TCP/IP, который благодаря простоте может использоваться как для коммутируемых, так и для выделенных каналов. SLIP поддерживается только протоколом сетевого уровня IP.

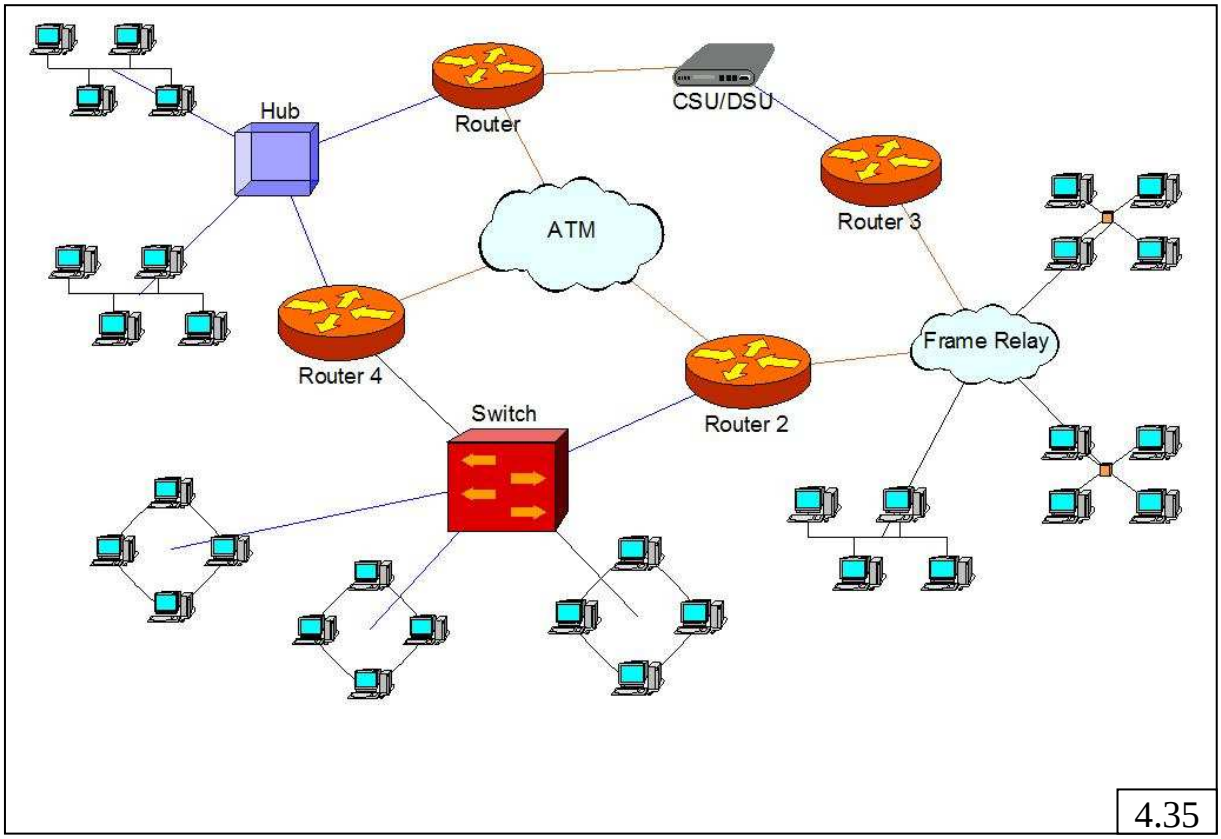
HDLC (High-level Data Link Control Procedure) – высокоуровневый протокол управления каналом – стандарт ISO для выделенных линий, представляющий собой семейство протоколов LAP (Link Access Protocol). HDLC относится к бит-ориентированным протоколам.

PPP (Point-to-Point Protocol) – протокол двухточечного соединения, пришедший на смену протоколу SLIP и построенный на основе формата кадров протоколов семейства HDLC с дополнением собственных полей. PPP является стандартным протоколом Интернета и так же, как протокол HDLC, представляет собой семейство протоколов.

4.4.3. Архитектурная концепция Internet

Структура сети Internet может быть представлена как множество компьютеров, называемых ***хостами***, подключенных к некоторой единой интерсети, представляющей собой совокупность физических сетей, называемых ***подсетями***, соединенных маршрутизаторами (рис.4.35). В

качестве подсетей могут выступать локальные сети, работающие под управлением некоторых аппаратно зависимых протоколов (Ethernet, Token Ring), или коммуникационные системы произвольной физической природы (модемные коммутируемые или выделенные линии, сети X.25, Frame Relay, FDDI, ATM и др.). При этом все функции протокола IP выполняют хосты и маршрутизаторы, называемые *узлами сети*.



4.35

Основным протоколом стека TCP/IP является протокол IP, который обеспечивает:

- *негарантированную доставку* пакетов, т.к. передаваемые по сети пакеты могут быть утеряны, дуплицированы, задержаны, доставлены с нарушением порядка;
- *дейтаграммную доставку без установления соединения*, то есть каждый пакет представляет собой обрабатываемый независимо от других блок данных, причем последовательно исходящие от отправителя пакеты могут распространяться по различным путям в сети, менять порядок и даже теряться;
- *максимально возможную доставку* пакетов в том смысле, что потеря пакета происходит лишь в той ситуации, когда протокол не находит никаких физических средств для его доставки.

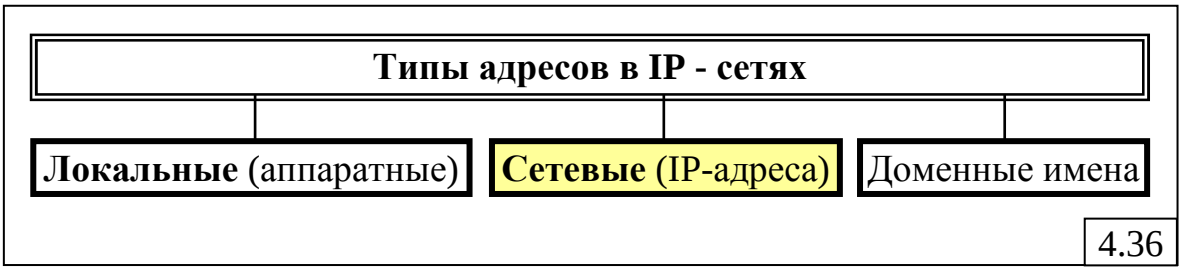
4.4.4. Адресация в IP-сетях

В стеке протоколов TCP/IP используются три типа адресов (рис.4.36):

- **физические (локальные) адреса**, используемые для адресации узлов в пределах подсети, например: MAC-адреса, если подсеть

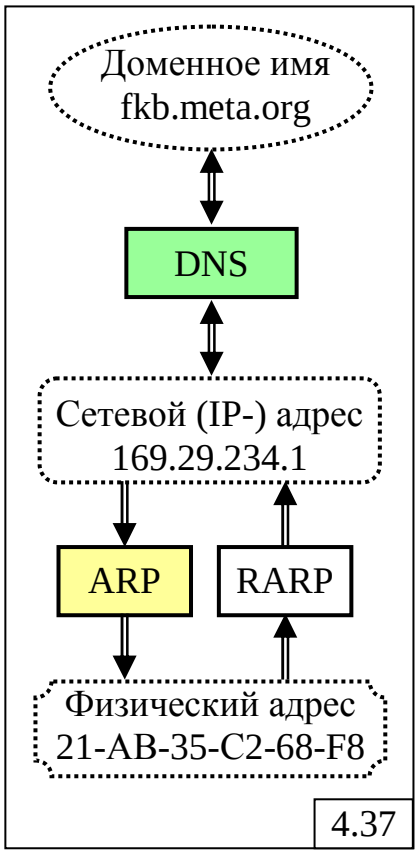
использует технологии Ethernet, Token Ring, FDDI, или IPX-адреса, если подсеть на основе технологии IPX/SPX;

- **сетевые (IP-адреса)**, используемые для идентификации узлов в пределах всей составной сети (подсети);
- **доменные имена** – символьные идентификаторы узлов, которыми оперируют пользователи.



4.4.4.1. Сетевые IP-адреса

Наличие трёх уровней адресации в IP-сетях требует применения процедур преобразования адресов разных уровней для установления соответствия между ними. Эти процедуры реализуются соответствующими протоколами, преобразующими адреса одного типа в другой.



Наиболее удобными для пользователей являются доменные имена, называемые также *доменными адресами*. Маршрутизация передаваемых данных в сети выполняется на основе сетевых адресов. В то же время, все устройства в компьютерной сети однозначно идентифицируются уникальными адресами канального уровня, в частности MAC-адресами в локальных сетях Ethernet и Token Ring.

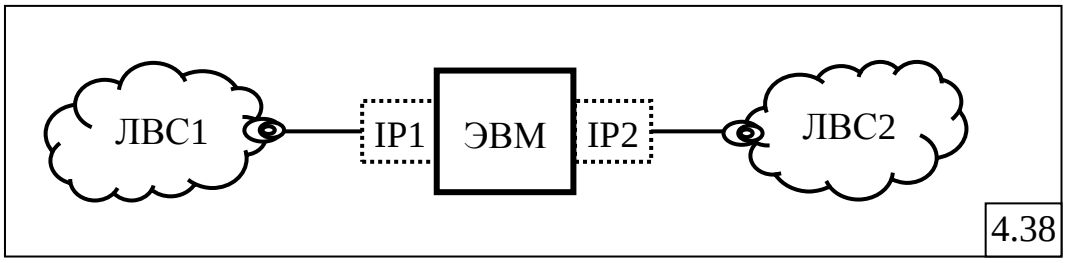
Преобразование адресов в IP-сетях осуществляется в соответствии со схемой, представленной на рис.4.37. Ниже подробно рассматриваются протоколы преобразования доменных адресов в сетевые и обратно с использованием протокола DNS и преобразование сетевых адресов в физические и обратно, реализуемое протоколами ARP и RARP соответственно.

4.4.4.2. Сетевые IP-адреса

IP-адрес – идентификатор *сетевого соединения (сетевого интерфейса)*. Это означает, что один и тот же компьютер, соединенный с двумя сетями (рис.4.38), имеет два IP-адреса: сеть 1 идентифицирует его по адресу IP1, а сеть 2 – по адресу IP2.

IP-адреса представляют собой 32-битовые идентификаторы, ориентированные на решение основной задачи протокола IP-

маршрутизации. Для удобства представления IP-адресов используется цифровое их написание в виде десятичного представления 4 байт, разделенных точками, например: **192.171.153.60** .



4.38

Первоначально в Интернете была принята так называемая классовая адресация. Все IP-адреса разделены на 5 классов (от A до E), представленных на рис.4.39, но практическое применение находят в основном три первых класса: A, B и C. Класс D предназначен для задания группового адреса, а класс E – не используется (зарезервирован для последующего использования).

Разряды	*1*	*2*	*3*	*4*	*5*	…	*9*	…	*17*	…	*25*	…	*32*
Класс A	0	Номер сети					Номер узла (хоста)						
Класс B	1	0	Номер сети						Номер узла				
Класс C	1	1	0	Номер сети							Номер узла		
Класс D	1	1	1	0	Групповой адрес								
Класс E	1	1	1	1	0	Зарезервирован для последующего использования							

4.39

IP-адрес состоит из двух полей: поле «Номер сети», представляющий собой адрес физической сети (подсети), и поле «Номер узла», выделяющий в этой подсети конкретное устройство (хост).

Признаком принадлежности адреса к определённому классу служат первые биты адреса: если первый бит равен 0, то адрес принадлежит классу A, если первый бит равен 1, а второй – 0, то адрес принадлежит классу B и т.д.

Принадлежность адреса к тому или иному классу определяет размер сети (табл.4.4):

- класс A соответствует большой сети с максимальным числом узлов $(2^{24} - 2) = 16\,777\,214$;
- класс B соответствует средней сети с числом узлов до 65534;
- класс C соответствует малой сети с числом узлов до $(2^8–2)=254$.

Отметим, что максимальное количество узлов в сети определяется количеством двоичных разрядов n, отводимых под номер узла: $N_{\max} = 2^n - 2$, то есть исключаются два номера:

- нулевой (все разряды равны 0); адрес с нулевым значением номера узла означает адрес сети;

- единичный (все разряды равны 1); адрес с единичными значениями номера узла является широковещательным и означает передачу пакета всем узлам сети.

Таблица 4.4

Показатель		Класс A	Класс B	Класс C
Размер сети		большая	средняя	малая
Номер (адрес) сети	наименьший	1.0.0.0	128.0.0.0	192.0.0.0
	наибольший	126.0.0.0	191.255.0.0	223.255.255.0
Число узлов в сети (max)		16 777 214	65 534	254
Длина поля в битах	номер сети	7	14	21
	номер узла	24	16	8

IP-адрес построен таким образом, чтобы поля «Номер сети» и «Номер узла» можно было бы выделить быстро, что особенно сказывается на эффективности маршрутизации (малые временные затраты на выделение адреса «Номер сети»).

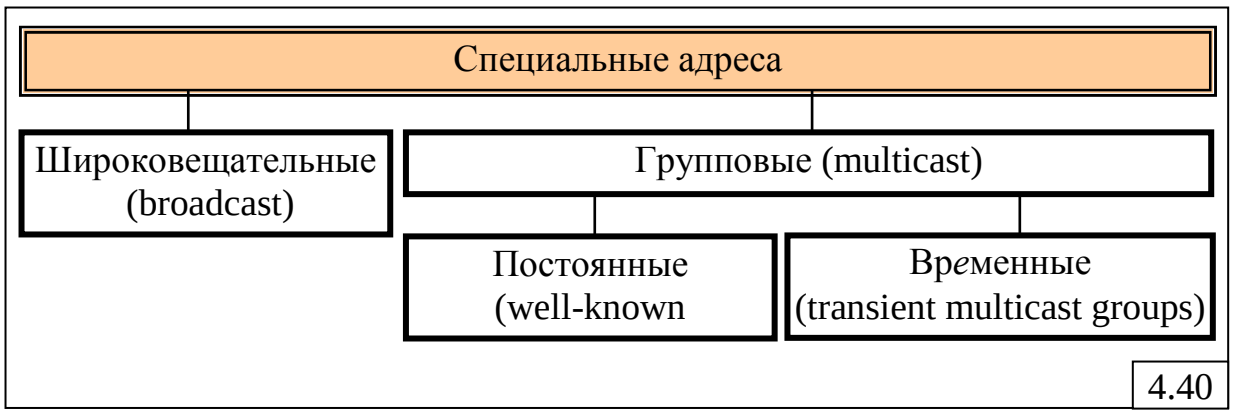
Поскольку IP-адрес идентифицирует сетевое соединение, а не узел, то отсюда вытекает принципиальное ограничение: *если компьютер переносится из одной подсети в другую, он должен обязательно изменить IP-адрес*.

4.4.4.3. Специальные, автономные и групповые IP-адреса

IP-адресация поддерживает **специальные адреса** (рис.4.40), обращенные к множеству узлов и/или сетей и делящиеся на два класса:

- широковещательные (broadcast), обращенные ко всем;
- групповые (multicast), обращенные к заданному множеству объектов.

Адресная нотация при этом основывается на заполнении адресных полей нулями (обращение к данному объекту, this) или единицами (обращение ко всем объектам, all).



4.40

В табл.4.5 перечислены все возможные специальные адреса, трактовка которых раскрывается ниже.

Таблица 4.5

Тип	Номер сети	Номер узла
Адрес 1	0 0 0 … 0	0 0 0 … 0
Адрес 2	0 0 0 … 0	X X X … X
Адрес 3	X X X … X	0 0 0 … 0
Адрес 4	1 1 1 … 1	1 1 1 … 1
Адрес 5	X X X … X	1 1 1 … 1
Адрес 6	01111111	- - - … -

Адрес 1 – "пустышка" или **неопределённый адрес**, используемый в инициализационной процедуре, когда рабочая станция не знает своего IP-адреса или хочет его согласовать; используется только как *адрес отправителя*, но никогда как адрес получателя.

Адрес 2 – адрес конкретного узла (XXX…X) в той же сети, что и узел-отправитель; применяется в случае, когда узел-отправитель не знает идентификатора сети, в которой работает, например при инициализации бездисковой рабочей станции, которая при включении вообще ничего не знает ни о сети, ни о себе; используется только как *адрес получателя* и никогда как адрес отправителя.

Адрес 3 – адрес сети (но не узла).

Адрес 4 – **локальный** или **ограниченный широковещательный** адрес (limited или local broadcast address); используется, когда идентификатор сети по каким-либо причинам неизвестен; для использования не рекомендуется.

Адрес 5 – ***прямой широковещательный*** адрес (direct broadcast address), обращенный ко всем узлам данной сети.

Адрес 6 – ***тестовый адрес***, в котором первый байт имеет значение 127, а оставшееся поле не специфицировано; используется для задач отладки и тестирования, не является адресом никакой сети, и маршрутизаторы никогда не обрабатывают его; также называется ***адресом обратной петли*** (*loopback* address), поскольку пакет с таким адресом, посланный на интерфейс loopback возвращается на тот же интерфейс, не выходя за пределы подсети.

Интерфейс loopback имеет несколько применений. Он может быть использован сетевым клиентским программным обеспечением, чтобы общаться с серверным приложением, расположенном на том же компьютере. Этот механизм полезен для тестирования служб, не подвергая их безопасность риску, как при удаленном сетевом доступе.

В стандартах Интернета определено несколько так называемых **автономных адресов**, рекомендуемых *для автономного использования в пределах одной подсети и необрабатываемых маршрутизаторами*:

- в классе A: 10.0.0.0 (1 сеть);

- в классе B: 172.16.0.0 – 172.31.0.0 (16 сетей);
- в классе C: 192.168.0.0 – 192.168.255.0 (256 сетей).

В качестве **группового адреса** используются адреса класса D. Групповая адресация в TCP/IP регламентируется входящим составной частью в IP протоколом *IGMP (Internet Group Management Protocol)*. Групповой адрес может объединять узлы из разных физических сетей путем использования в маршрутизаторах специальных протоколов групповой маршрутизации. Каждый узел может в любой момент подключиться к определенной адресной группе или выйти из нее.

Групповые адреса назначаются NIC и разделяются на два класса:

- **постоянные** – для непрерывно существующих групп (так называемые «всем известные адреса» – well-known addresses);
- **вр*е*менные** – для организуемых на некоторый срок групп, которые существуют до тех пор, пока в группе сохраняется хотя бы один член (так называемые «временные адресные группы» – transient multicast groups).

Распространение групповых сообщений по интерсети ограничивается временем жизни (time-to-live) IP-пакета.

4.4.4.4. Использование масок для IP-адресов

Маска представляет собой 32-разрядный двоичный код, содержащий в *нескольких первых (старших) разрядах* «единицы», а в остальных – «нули». Количество единиц в маске определяет границу номера (идентификатора) сети. Другими словами, единичные значения маски позволяют выделить из IP-адреса номер сети, а оставшиеся младшие разряды IP-адреса определяют номер узла в этой сети.

Использование масок для IP-адресов позволяет *расширить адресное пространство* и сделать систему адресации более гибкой, не привязанной к классам IP-адресов **(**A, B или C).

Пример. Пусть заданы:

IP-адрес: **126.65.32.5** и *маска:* **255.192.0.0** .

IP-адрес **126.65.32.5** соответствует адресу узла ***0.65.32.5*** в сети ***126.0.0.0*** .

Запишем IP-адрес и маску в двоичном виде:

IP-адрес:	**01111110.01**	000001.00100000.00000101
маска:	**11111111.11**	000000.00000000.00000000

Тогда:

адрес сети: **01111110.01** или **126.64.0.0**

адрес узла: 000001.00100000.00000101 или **0.1.32.5**

Таким образом, вместо сети 126.0.0.0, принадлежащей к классу A, при наличии маски имеем сеть 126.64.0.0, которая не принадлежит ни одному из классов A, B или C. Максимальное количество узлов в этой сети определяется длиной поля адреса, используемого для нумерации узлов, то

есть количеством нулевых разрядов в маске. В нашем примере это 22 разряда, следовательно, максимальное количество узлов в сети будет равно $2^{22} – 2 = 4\ 194\ 302$.

Для стандартных классов сетей маски имеют вид:

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).

Часто использование маски указывается в виде: 116.165.42.35/**12**, где число 12 определяет количество единичных разрядов в маске для IP-адреса 116.165.42.35.

4.4.4.5. Распределение IP-адресов

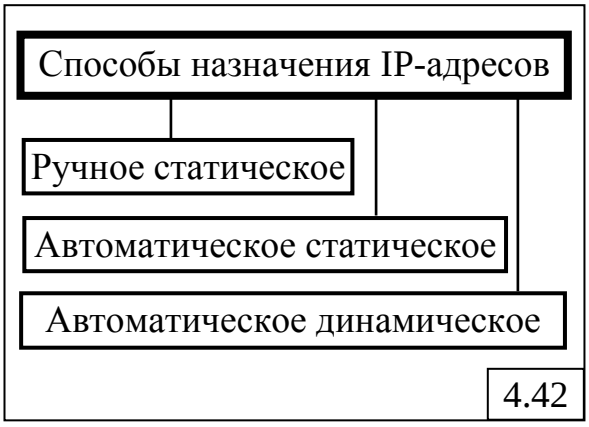
Распределение IP-адресов может выполняться двумя способами:

- *централизованное распределение,* реализуемое специальными органами регистрации глобальных адресов, распределяющими адреса в сети Интернет и образующими иерархическую структуру, показанную на рис.4.41;
- *автоматизированное распределение,* реализуемое в сетях с единым административным управлением с использованием протокола назначения адресов DHCP.

Протокол для автоматического назначения IP-адресов – **Dynamic Host Configuration Protocol (DHCP)** – может поддерживать следующие способы распределения адресов (рис.4.42):

- *ручное распределение* – с участием администратора сети, причем DHCP-сервер всегда выдает определенному клиенту один и тот же назначенный ему администратором адрес;
- *автоматическое статическое распределение* – DHCP-сервер при первом подключении клиента выбирает из пула наличных IP-адресов произвольный IP-адрес, который при последующих подключениях клиента не меняется;

- *автоматическое динамическое распределение* – DHCP-сервер при каждом обращении клиента выдает IP-адрес на ограниченное время – **время аренды**, причем впоследствии этот адрес может быть предоставлен другому компьютеру; это позволяет строить IP-сеть с числом узлов, превышающим количество имеющихся в распоряжении администратора IP-адресов.



Кроме IP-адреса DHCP-сервер может назначить клиенту другие параметры стека TCP/IP, например:

- маску;
- IP-адрес маршрутизатора по умолчанию;
- IP-адрес сервера DNS;
- доменное имя компьютера и т.п.

Постоянный рост сети Интернет ведет к **дефициту IP-адресов**, особенно адресов класса А. Кроме того, имеющееся в распоряжении некоторой сети адресное пространство часто используется нерационально, например, используются не все адреса из 254 имеющихся в распоряжении сети класса С.

4.4.4.6. Бесклассовая междоменная маршрутизация

Использование масок переменной длины для IP-адресов позволяет не только расширить адресное пространство за счет увеличения количества номеров сетей, но и экономно выделять IP-адреса.

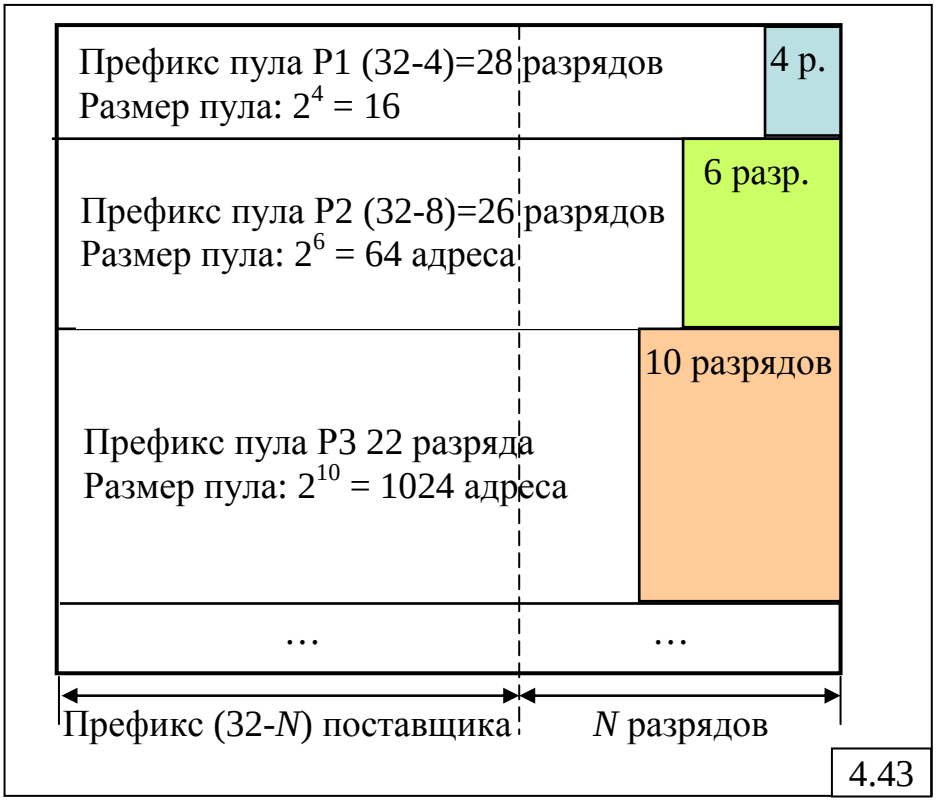
Например, если в какой-то небольшой сети находится десяток узлов, то очевидно, что неразумно выделять ей номер сети даже класса С, обеспечивающей нумерацию 254-х узлов. Гораздо более эффективным будет выделение для этой сети небольшого количества IP-адресов.

Для выделения ограниченного количества IP-адресов разработана технология ***бесклассовой междоменной маршрутизации*** **(CIDR** – Classless Inter-Domain Routing), использующая *бесклассовую адресацию* и позволяющая гибко распределять IP-адреса.

Для реализации технологии CIDR необходимо, чтобы организация, распределяющая IP-адреса, имела в наличии непрерывный диапазон адресов. Это предоставляет возможность выделять сетям некоторое количество IP-адресов, имеющих *одинаковый префикс*, то есть одинаковые значения в нескольких старших разрядах. На рис.4.43 показан пример, иллюстрирующий принцип выделения адресов из общего пула адресов для сетей разных размеров. Так, например:

- пул P1 имеет префикс длиной 28 двоичных разрядов и 4 разряда под нумерацию узлов, что позволяет пронумеровать 16 узлов небольшой сети;
- пул P2 имеет префикс длиной 26 разрядов и 6 разрядов под нумерацию узлов, что позволяет пронумеровать 64 узла;

- пул P3 с префиксом длиной 26 разрядов позволяет пронумеровать 1024 узла.



При таком выделении адресов необходимо, чтобы:

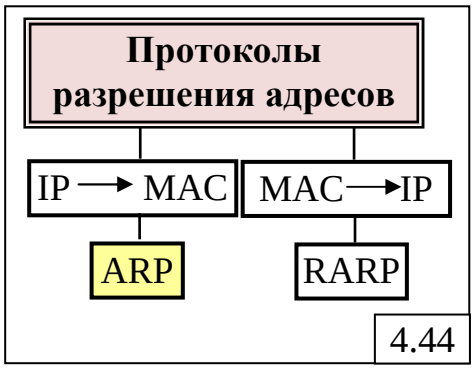
- количество выделяемых адресов было кратно степени двойки;
- начальная граница выделяемого пула адресов была кратна требуемому количеству узлов.

Благодаря технологии CIDR имеется возможность нарезать блоки адресов в соответствии с действительными потребностями каждой сети.

4.4.4.7. Протоколы разрешения адресов ARP и RARP

Определение физического адреса устройства (MAC-адреса) по его IP-адресу и наоборот, IP-адреса по MAC-адресу, решают входящие в IP-стек два протокола:

- **ARP** (Address Resolution Protocol – Протокол разрешения адресов)
- **RARP** (Reverse Address Resolution Protocol – Протокол обратного определения адреса) соответственно рис.4.44.



Протокол ARP поддерживает в каждом узле (сетевом адаптере или порту маршрутизатора) ***ARP-таблицу***, содержащую (рис.4.45):

- IP-адрес;
- MAC-адрес;
- тип записи (динамический, статический).

IP-адрес	MAC-адрес	Тип записи
195.36.210.12	12-43-F4-AB-5C-01	Динамический/статический

4.45

По этой таблице узел может определить физический адрес (MAC-адрес) узла назначения, находящегося *в этой же сети*, по известному IP-адресу и указать его в заголовке кадра канального уровня. Если в ARP-таблице отсутствует запись для некоторого IP-адреса, то узел формирует *широковещательное* сообщение – ***ARP-запрос***, в котором запрашивает физический адрес узла назначения. Все узлы сети принимают этот запрос, однако лишь один узел, IP-адрес которого совпадает с указанным в ARP-запросе, отвечает на него, высылая ***ARP-ответ*** со своим физическим адресом *непосредственно узлу*, приславшему ARP-запрос. Последний записывает в ARP- таблицу найденное соответствие между IP-адресом и MAC-адресом и в дальнейшем не запрашивает его при повторных обращениях к этому узлу. Протокол ARP предполагает, что узлы знают свои IP-адреса.

Формат ARP-запроса (ответа) представлен на рис.4.46.

Поле	Значение
«Тип сети» канального уровня	**1** (для Ethernet)
«Тип протокола» сетевого уровня	**2048** (=0800_{16} для IP)
«Длина локального адреса»	**6** (для Ethernet)
«Длина сетевого адреса»	**4** (для IP)
«Опция»	(**1** – для ARP-запроса и **2** – ответа)
«Локальный адрес отправителя»	008048EB6A15
«Сетевой адрес отправителя»	195.67.8.9
«Локальный адрес получателя»	**000000000000** (для ARP-запроса)
«Сетевой адрес получателя»	195.67.8.12

4.46

В сети, объединяющей несколько локальных сетей (подсетей) с помощью маршрутизаторов, продвижение пакетов от узла, находящегося в одной подсети, к узлу, находящемуся в другой подсети, осуществляется на основе старшей части IP-адреса, то есть на основе номера сети. После того, как пакет поступит в конечный маршрутизатор, к которому подсоединена вторая подсеть (сеть назначения), необходимо этот пакет упаковать в кадр и в качестве физического адреса узла назначения указать его MAC-адрес. Маршрутизатор просматривает свою ARP-таблицу и, если не находит соответствующего IP-адреса, формирует широковещательный ARP-запрос, посылает его в локальную сеть и ожидает ARP-ответа. Если в сети нет

компьютера с указанным в ARP-запросе IP-адресом, то ARP-ответа не будет, и протокол IP уничтожит все пакеты, направляемые по этому адресу.

Статические записи создаются вручную и существуют, пока соответствующий узел (компьютер или маршрутизатор) не будет выключен.

Динамические записи создаются протоколом ARP как по собственным ARP-запросам, так и путем извлечения из широковещательных запросов IP- и MAC-адресов отправителя. Динамические записи периодически обновляются. Если в течение определенного интервала времени (порядка нескольких минут) адрес не использовался, то он исключается из таблицы.

В глобальных сетях, не поддерживающих широковещательные сообщения, ARP-таблицы формируются администратором вручную и помещаются на какой-либо хост, либо выделяется специальный маршрутизатор, который автоматически ведет ARP-таблицу для всех остальных узлов этой автономной сети.

Протокол RARP используется в случае, если узел – бездисковая рабочая станция, у которой только что включили питание и она не только ничего не знает о себе и окружающих, но и не может произвести дистанционную загрузку операционной системы, которая хранится на сетевом диске.

Узел широковещательно вызывает обслуживающий его сервер, закладывая в запрос свой физический адрес (при этом узел может даже не знать адреса сервера). В сети находится по меньшей мере один обслуживающий такие запросы сервер (RARP-сервер), который распознает запрос от рабочей станции, выбирает из некоторого списка свободный IP-адрес и шлет этому узлу сообщение с необходимой информацией:

- динамически выделенный узлу IP-адрес;
- свой физический адрес;
- IP-адрес и т.д.

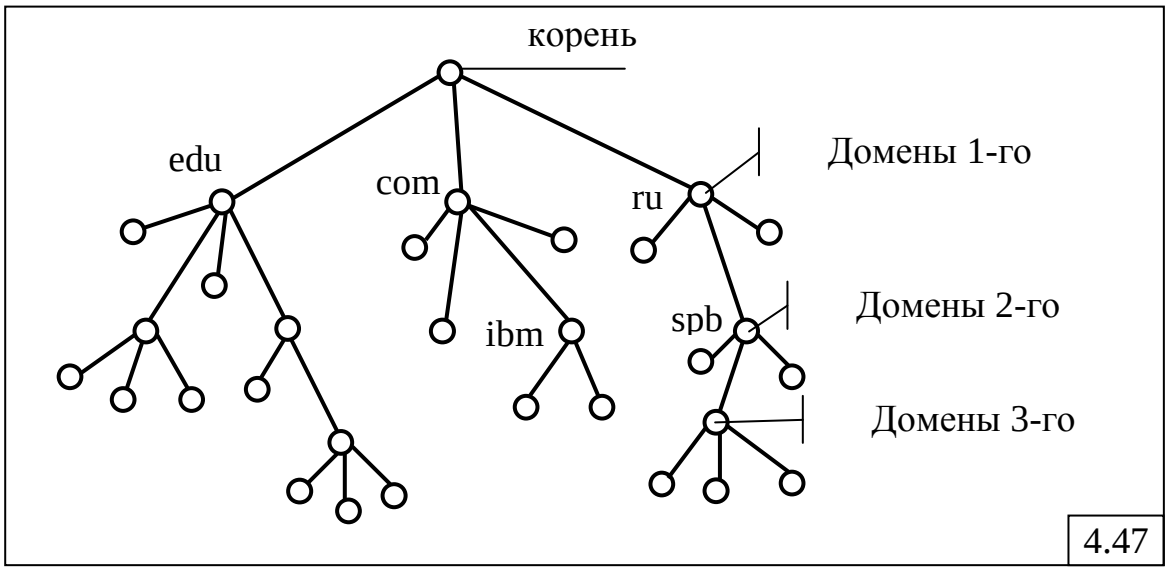
Поскольку при таком механизме отказ RARP-сервера очень критичен в том смысле, что без его услуг не заработает целый ряд рабочих станций, то обычно сеть конфигурируется так, чтобы протокол RARP поддерживало несколько серверов в сети.

4.4.4.8. Система доменных имен DNS

Доменное имя – символьное имя компьютера.

В стеке TCP/IP применяется система доменных имен с *иерархической древовидной структурой* (рис.4.47), допускающей использование в имени произвольного количества составных частей.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют ***домен (domain) имен***.



4.47

Примерами доменных имён организаций являются:

- com – коммерческие организации;
- edu – образовательные организации;
- gov – правительственные организации;
- org – некоммерческие организации;
- net –организации поддержки сетей.

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального узла, так и средствами централизованной службы, реализуемой системой доменных имён.

Система доменных имен (Domain Name System – DNS) – централизованная служба, основанная на распределенной базе отображений «доменное имя – IP-адрес» (рис.4.48).

Служба DNS использует в своей работе протокол типа «клиент-сервер», в котором определены такие понятия как ***DNS-сервер***, поддерживающий распределенную базу отображений, и ***DNS-клиент***, обращающийся к DNS-серверу с запросом. DNS-сервер использует текстовые файлы формата «IP-адрес – доменное имя».

Доменное имя	IP-адрес
sota.park.org	213.45.7.12
abc.spb.ru	184.31.61.1
labor.uni.edu	159.1.26.34

4.48

Служба DNS является распределенной. Каждый DNS-сервер хранит имена следующего уровня иерархии и кроме таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов, что упрощает процедуру поиска.

Для ускорения поиска IP-адресов в DNS-серверах применяется процедура кэширования проходящих через них ответов на определенное время – от нескольких часов до нескольких дней.

4.4.5. Коммуникационный протокол IPv4

Протокол IP специфицирует три основных элемента:

- блок данных – **пакет IP**, с которым работает протокол;
- механизмы распространения (маршрутизации) пакетов;
- способы обработки конфликтных ситуаций.

Пакет IP состоит из заголовка и блока данных (рис.4.49,а).

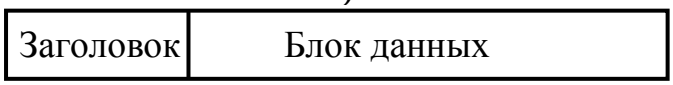
В настоящее время в сети Интернет могут циркулировать IP-пакеты двух версий:

- IP-пакет версии 4 (IPv4);
- IP-пакет версии 6 (IPv6).

Протокол IP обрабатывает и интерпретирует только поля заголовка.

Формат заголовка пакета IPv4 показан на рис.4.49,б).

а)



б)

Биты: 0 … 3	4 … 7	8 … 15	16 … 23	24 … 31
Номер версии	Длина заголовка	Тип сервиса: PR D T R --	Общая длина	
Идентификатор пакета			Флаги: - DF MF	Смещение фрагмента
Время жизни		Протокол	Контрольная сумма заголовка	
IP-адрес источника				
IP-адрес назначения				
Параметры				Наполнение

4.49

Рассмотрим назначения полей заголовка.

«Номер версии» (4 бита) – используется для указания версии протокола IP, который должен обрабатывать данный пакет. В настоящее время осуществляется постепенный переход от версии 4 к версии 6, и большинство узлов могут обрабатывать пакеты обеих версий. Если это поле содержит значение, отличное от указанных версий протокола, пакет уничтожается.

«Длина заголовка» (4 бита) – задает значение длины заголовка пакета, измеренной в 32-битовых (4-байтовых) словах. Минимальное значение длины (при отсутствии необязательных полей «Параметры» и «Наполнение») равно 5, что соответствует заголовку длиной 20 байт. Максимальное значение этого 4-битового поля равно 15, что соответствует

заголовку длиной 60 байт. Следовательно, максимальный размер необязательных полей «Параметры» и «Наполнение» равен 40 байтам.

«Тип сервиса» (Type of Service, ToS) – 8-битовое поле, предназначенное для оптимизации транспортной службы, содержащее:

- 3-битовое поле «Приоритет» принимает 8 значений: от 0 (нормальный приоритет) до 7 (сетевое управление);
- биты D,T,R задают тип транспортировки, который "запрашивает" пакет; установка этих битов в состояние "1" требует:
  - D=1 (Delay – *задержка*) – малой задержки при передаче пакета;
  - T=1 (Throughput – *пропускная способность*) – высокой пропускной способности;
  - R=1 (Reliability – *надежность*, достоверность) – высокой надежности;
- 2 резервных бита.

Стандарты, принятые в конце 90-х годов, дали новое название этому полю – ***байт дифференцированное обслуживание*** или ***DS-байт*** – и переопределили назначение его битов.

Поле «Тип сервиса» не всегда используется маршрутизаторами.

«Общая длина» (16 бит) – задает длину пакета, включая заголовок и данные, измеренную в байтах. Общая длина пакета IP может достигать 65 535 байт, однако в большинстве сетей столь большие пакеты не используются.

Протокол IP должен обеспечивать межсетевое взаимодействие между разными сетями, различающимися, в том числе, ограничением на *максимальную длину кадра*, разрешенным в той или иной физической сети (Maximum Transfer Unit, MTU). Поэтому протокол IP вынужден решать задачу, более свойственную транспортному протоколу, – разбивку больших пакетов на малые и наоборот – их сборку. Это требуется делать в тех случаях, когда на вход некоторой физической сети поступает пакет, превосходящий по длине MTU для данной сети. Такая операция называется ***фрагментированием*** (fragmentation) и осуществляется следующим образом.

Блок данных большого исходного пакета разделяется на ***фрагменты*** длиной MTU для физической сети, в которую направляются фрагменты. При этом фрагменты упаковываются в пакеты, заголовки которых похожи на заголовок исходного пакета.

В стандартах TCP/IP предусматривается, что все узлы должны принимать пакеты длиной не менее 576 байт, независимо от того, являются они фрагментами или целыми пакетами.

Следующие три поля заголовка пакета указывают на то, что данные пакеты являются фрагментами одного большого пакета.

«Идентификатор пакета» (16 бит) – общий для всех фрагментов идентификатор, указывающий на принадлежность фрагмента к одному большому пакету.

«Флаги» (3 бита) – содержат признаки (биты), связанные с фрагментацией:

- DF (Do not Fragment – не фрагментировать) – значение, равное 1, запрещает маршрутизатору фрагментировать пакет;
- MF (More Fragments – больше фрагментов) – значение, равное 1, означает, что фрагмент является промежуточным;
- один бит зарезервирован.

«Смещение фрагмента» (13 бит) – смещение *в байтах* поля данных этого фрагмента относительно начала поля данных исходного нефрагментированного пакета. Смещение используется при сборке фрагментов в пакет и должно быть кратно 8 байтам.

«Время жизни» (Time To Live, TTL) – 8-битовое поле, содержащее время, измеряемое в секундах, в течение которого пакет может существовать в сети. Хосты и маршрутизаторы, обрабатывающие данный пакет, уменьшают значение этого поля в период обработки и хранения пакета как минимум на 1 плюс время ожидания в очереди. Однако на практике в каждом маршрутизаторе обычно из этого времени просто вычитается 1. Таким образом, время жизни фактически измеряется количеством маршрутизаторов, через которые проходит пакет. Когда время жизни истекает, пакет уничтожается. При этом источник сообщения уведомляется о потере пакета. Наличие конечного времени жизни пакета, равное 255 (8 двоичных разрядов), обеспечивает, в частности, защиту от таких нежелательных событий, как передача пакета по циклическому маршруту, перегрузка сетей.

«Протокол» (8 бит) – указывает протокол вышележащего уровня, которому предназначена информация, содержащаяся в поле данных пакета IP. Например, значение 6 соответствует протоколу TCP, а значение 17 – протоколу UDP.

«Контрольная сумма заголовка» (16 бит) – используется для контроля целостности только заголовка пакета IP и вычисляется как сумма всех 16-битовых полуслов заголовка в дополнительном коде, преобразованная также в дополнительный код. Таким образом, вычисляемая получателем контрольная сумма заголовка вместе с этим полем должна быть равна нулю. Поскольку некоторые поля заголовка могут изменять свои значения в процессе передачи пакета по сети, контрольная сумма вычисляется и проверяется в каждом маршрутизаторе и в конечном узле.

«IP-адрес источника» (32 бита) – IP-адрес отправителя пакета.

«IP-адрес назначения» (32 бита) – IP-адрес получателя пакета.

«Параметры» – необязательное поле переменной длины, применяемое для указания параметров, используемых обычно при отладке сети и связанных, например, с режимами безопасности или маршрутизации.

«Наполнение» – поле переменной длины, необходимое для дополнения заголовка пакета до целого числа 32-битовых слов.

4.4.6. Коммуникационный протокол IPv6

Проблемы, с которыми в начале 90-х годов столкнулись разработчики и пользователи Интернета, базирующегося на протоколах TCP/IP, привели к осознанию необходимости разработки новой версии протокола IP – ***протокола IPv6***, который должен обеспечить достижение следующих целей:

- создание масштабируемой системы адресации, обеспечивающей поддержку миллиардов хостов даже при неэффективном использовании адресного пространства;
- уменьшение таблиц маршрутизации и упрощение протокола для ускорения обработки пакетов маршрутизаторами;
- предоставление гарантий качества транспортных услуг при передаче неоднородного трафика, в частности, при передаче данных реального времени;
- более надёжное обеспечение безопасности - аутентификации и конфиденциальности;
- возможность сосуществования старого и нового протоколов;
- возможность развития протокола в будущем.

Основными особенностями протокола IPv6 являются следующие.

1. Длина IP-адреса увеличена до 16 байт, что предоставляет пользователям практически неограниченное адресное пространство

2. Упрощена структура заголовка, содержащего всего 7 полей (вместо 13 в протоколе Ipv4), что позволяет маршрутизаторам быстрее обрабатывать пакеты, то есть повышает их производительность.

3. Улучшена поддержка необязательных параметров, так как в новом заголовке требуемые прежде поля стали необязательными, а изменённый способ представления необязательных параметров ускоряет обработку пакетов в маршрутизаторах за счёт пропуска не относящихся к ним параметров.

4. Улучшена система безопасности.

5. Предусмотрена возможность расширения типов (классов) предоставляемых услуг, которые могут появиться в результате ожидаемого роста мультимедийного трафика.

4.4.6.1. Адресация в IPv6

Необходимость расширения адресного пространства в сетях TCP/IP была одной из основных целей перехода на новую версию протокола IP. Для этого длина IP-адреса была увеличена до **16 байт** или **128 бит**, что предоставляет пользователям практически *бесконечное адресное пространство* – более чем 10^{38} адресов.

В протоколе IPv6 вместо двухуровневой (как в IPv4) иерархии адресов используется четырёхуровневая:

- 3 уровня используются для идентификации сетей;
- 1 уровень используется для идентификации узла сети.

Для записи 16-байтовых адресов используется *шестнадцатеричная форма* (вместо десятичной формы в протоколе IPv4), причём каждые 4 шестнадцатеричные цифры отделяются друг от друга *двоеточием*:

AB25:164:0:E12B:6:0:C2C4:1234

BDA5::3217:19:0:F084 .

Как видно из представленных примеров, при записи адреса допускается ряд упрощений:

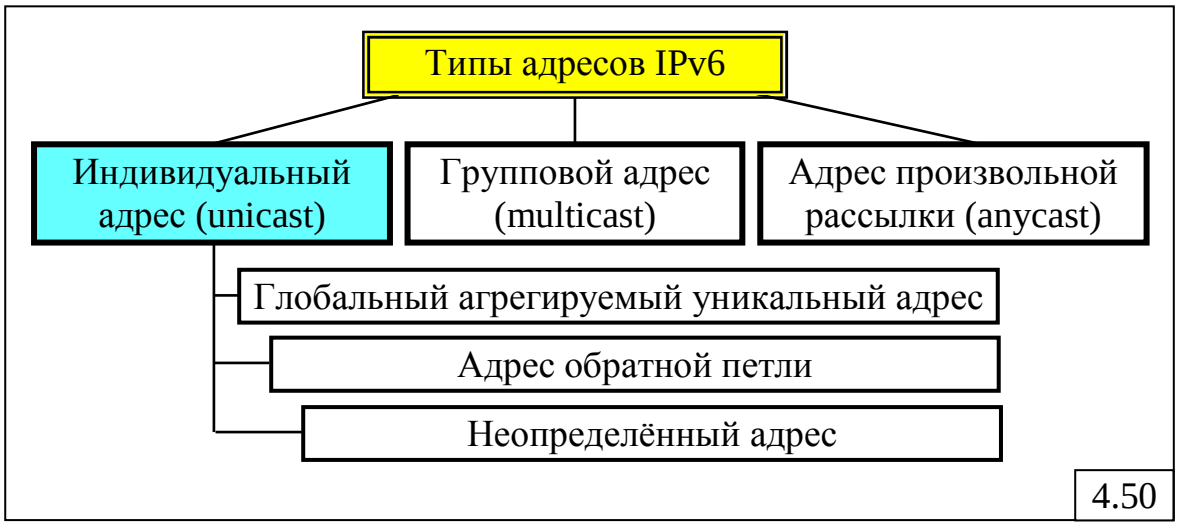
- вместо 4-х нулей записывается только один нуль: **0** вместо 0000;
- можно опускать незначащие нули в начале каждого четырёхсимвольного поля адреса: **164** вместо 0164 или **6** вместо 0006;
- если в адресе имеется длинная последовательность нулей, то запись можно сократить, заменив в ней все нули двоеточием, причём двоеточие может употребляться только один раз:

CF18: 35::67:5 , что соответствует адресу **CF18: 35:0:0:0:0:67:5** ;

- для сетей, использующих обе версии (IPv4 и IPv6) протокола разрешается использовать традиционную десятичную запись IPv4 в 4-х младших байтах, например: **::BAC2:192.85.1.6** .

В протоколе IPv6 предусмотрено 3 типа IP-адресов (рис.4.50):

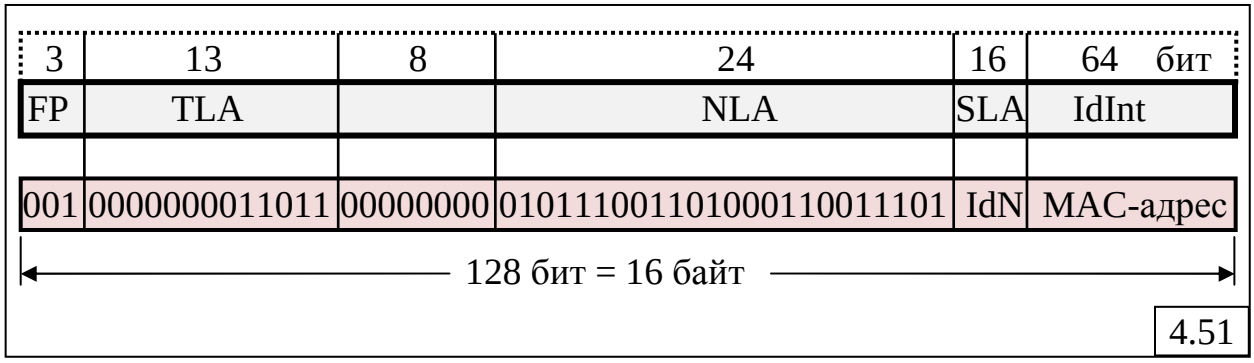
- ***индивидуальный адрес*** (unicast), определяющий уникальный идентификатор отдельного интерфейса оконечного узла или маршрутизатора;
- ***групповой адрес*** (multicast), аналогичный групповому адресу IPv4, идентифицирует группу интерфейсов, относящихся, как правило, к разным узлам;
- ***адрес произвольной рассылки*** (anycast) – новый тип адреса, назначаемый только интерфейсам маршрутизатора и определяющий группу интерфейсов, к одному из которых доставляется пакет с таким адресом, как правило, «ближайшему» в соответствии с метрикой, используемой протоколами маршрутизации.



Индивидуальные IP-адреса могут быть трёх типов (рис.4.50):

- ***глобальный агрегируемый уникальный адрес***, являющийся основным подтипом индивидуального адреса, основанные на агрегировании для упрощения маршрутизации;
- ***адрес обратной петли***, играющий ту же роль, что и адрес 127.0.0.1 протокола IPv4 и имеющий вид: **0:0:0:0:0:0:0:1**;
- ***неопределённый адрес***, состоящий из одних нулей и являющийся аналогом адреса 0.0.0.0 протокола IPv4.

Рассмотрим структуру глобального агрегируемого уникального адреса (рис.4.51).



4.51

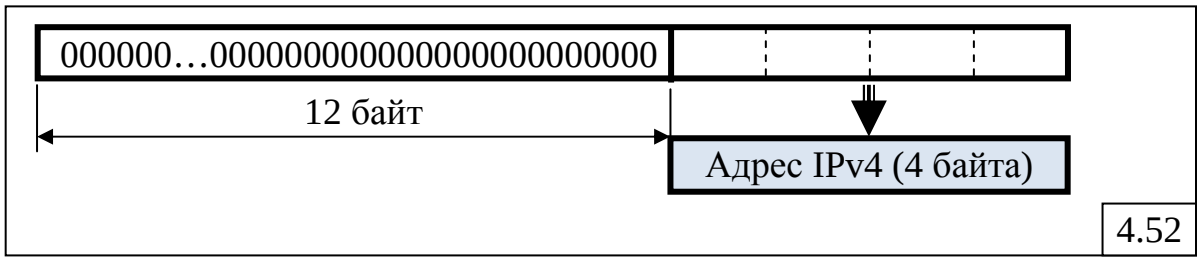
Поле **FP** (Format Prefix – **префикс формата)** определяет формат адреса и для рассматриваемого типа имеет значение 001.

Следующие поля описывают три уровня идентификации сетей:

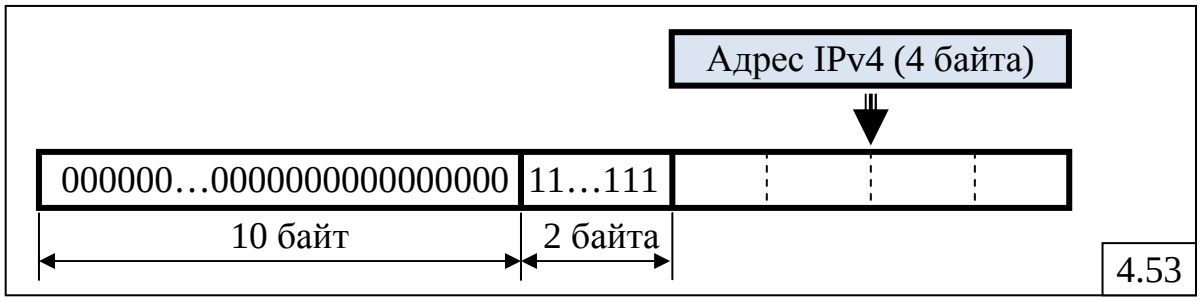
- **TLA** (Top-Level Aggregation – **агрегирование верхнего уровня**) предназначено для нумерации сетей самых крупных поставщиков услуг; небольшое количество разрядов (13 двоичных разрядов) позволяют ограничить количество таких сетей числом 8196 и, следовательно, ограничить размер таблиц маршрутизации и ускорить работу магистральных маршрутизаторов; следующие 8 разрядов за полем TLA зарезервированы на будущее для его расширения;
- **NLA** – (Next-Level Aggregation – **агрегирование следующего уровня**) предназначено для нумерации средних и мелких поставщиков услуг;
- **SLA** – (Site-Level Aggregation – **агрегирование местного уровня**) предназначено для нумерации подсетей, находящихся в распоряжении одного администратора, который может формировать адреса, состоящие из идентификатора подсети SLA и идентификатора интерфейса IdInt, без согласования с поставщиком услуг.

Поле **IdInt** – идентификатор интерфейса является аналогом номера узла в протоколе IPv4, но в отличие от него содержит физический (локальный) адрес интерфейса (например, MAC-адрес или адрес X.25), а не произвольно назначенный номер узла. В этом случае *отпадает необходимость в протоколе ARP* и в ручном конфигурировании конечных узлов. Кроме того, *теряет смысл использование масок для разделения сетей на подсети*, в то время как объединение сетей приобретает особое значение.

Для того чтобы узлы, поддерживающие протокол IPv6, могли передавать пакеты через сеть IPv4, разработан специальный подтип адресов, которые переносят адрес IPv4 в младших 4-х байтах адреса IPv6, а в 12 старших байтах содержат нули (рис.4.52).

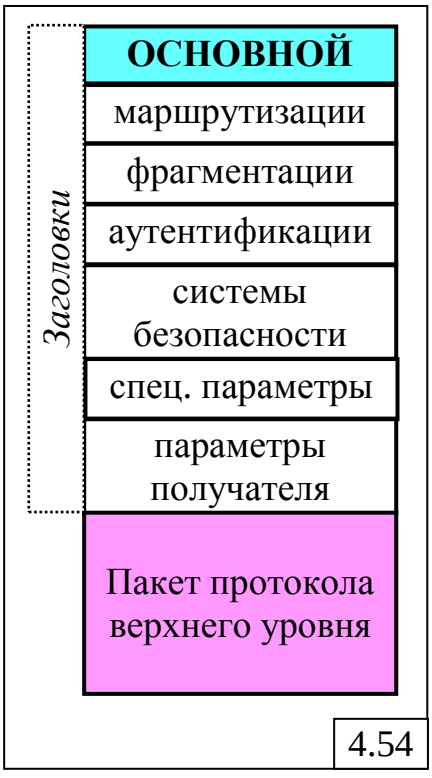


Для передачи пакетов IPv4 через подсети, работающие по протоколу IPv6 предназначен **IPv4-отображённый IPv6-адрес** (рис.4.53), содержащий в первых десяти байтах нули, а в двух последующих байтах – единицы, которые показывают, что данный узел поддерживает только протокол IPv4.



4.4.6.2. Структура пакета IPv6

Структура пакета IPv6 (рис.4.54) существенно отличается от пакета IPv4. Это проявляется, прежде всего, в возможности наличия нескольких заголовков – кроме основного заголовка, который всегда присутствует, пакет может иметь несколько дополнительных заголовков, которые могут содержать информацию, необходимую для качественной передачи пакета.



В качестве дополнительных заголовков могут использоваться следующие:

- **заголовок маршрутизации**, содержащий полный маршрут при маршрутизации от источника;
- **заголовок фрагментации**, содержащий информацию о фрагментации исходного IP-пакета;
- **заголовок аутентификации**, содержащий информацию, необходимую для аутентификации конечных узлов и обеспечения целостности содержимого IP-пакетов;

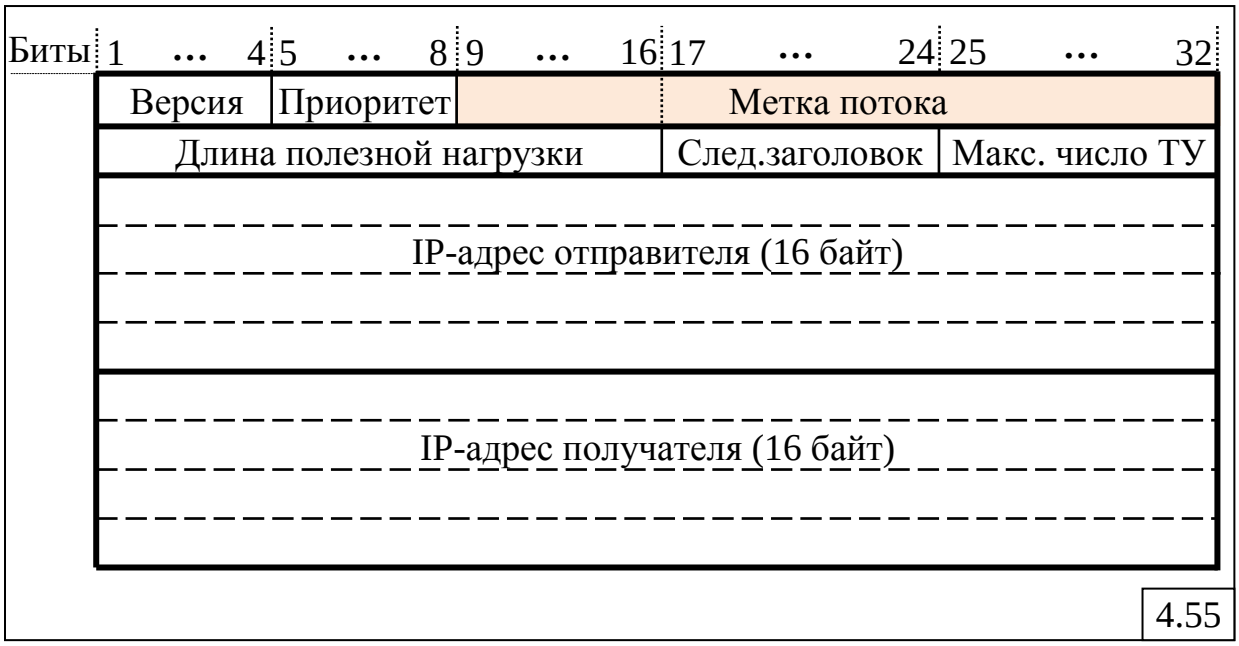
- **заголовок системы безопасност**и, содержащий информацию, необходимую для обеспечения конфиденциальности передаваемых данных путём шифрования пакетов;
- **специальные параметры**, необходимые для обработки пакетов в процессе передачи по сети;
- **параметры получателя**, содержащие дополнительную информацию для узла назначения.

Такая структура пакета IPv6 обеспечивает следующие преимущества по сравнению с пакетом IPv4:

- *меньше нагрузка на маршрутизаторы*, поскольку все дополнительные заголовки обрабатываются только в конечных узлах;
- *большая функциональность и открытость* для внедрения новых механизмов протокола IP за счёт использования большого количества дополнительных параметров.

4.4.6.3. Формат основного заголовка IPv6

Формат основного заголовка IPv6 имеет фиксированную длину 40 байт (рис.4.55).



Поле «Версия» (4 бита) содержит число 6 для пакета IPv6.

Поле «Приоритет» (4 бита) используется для того, чтобы различать пакеты с разными требованиями к доставке в реальном времени.

Поле «Метка потока» предназначено для установления между отправителем и получателем псевдосоединения с определёнными свойствами и требованиями. Маршрутизаторы, в зависимости от метки потока в заголовке прибывшего пакета, определяют, какого рода особая обработка требуется пакету. С помощью этого поля протокол пытается объединить достоинства дейтаграммного способа передачи пакетов и способа «виртуальный канал».

Поле «Длина полезной нагрузки» указывает, сколько байт содержится в пакете без учета основного заголовка, длиной 40 байт. Аналогичное поле «Полная длина» в IPv4 определяло всю длину пакета с учётом заголовка.

Поле «Следующий заголовок» указывает, какой из дополнительных заголовков следует за основным. Все дополнительные заголовки содержат такие же поля, которые указывают на последующие заголовки. В последнем заголовке в этом поле указывается протокол транспортного уровня (TCP или UDP), которому следует передать содержимое пакета.

Поле «Максимальное число транзитных участков (ТУ)» определяет время жизни пакета. Значение поля, устанавливаемое узлом-отправителем, уменьшается на единицу на каждом транзитном участке.

Далее следуют 16-байтные IP-адреса отправителя и получателя.

Сравнение заголовка IPv6 с заголовком IPv4 показывает, что:

- поле «Длина заголовка» исчезло, так как основной заголовок IPv6 имеет фиксированную длину;
- поле «Протокол» отсутствует, поскольку поле «Следующий заголовок» указывает, что следует за последним заголовком (TCP-сегмент или UDP-пакет);
- удалены поля, относящиеся к фрагментации, так как все узлы, поддерживающие протокол IPv6, должны динамически определять нужный размер дейтаграммы, что делает фрагментацию маловероятной;
- минимальный размер пакета, который должен передаваться в сетях IPv6 без фрагментации, увеличен с 576 до 1280 байт;
- поле «Контрольная сумма» удалено, так как её подсчёт занимает много времени, что существенно снижает производительность узлов; к тому же всё шире используются надёжные линии связи, например волоконно-оптические.

Таким образом, протокол IPv6 является *простым*, *быстрым* и *гибким* протоколом сетевого уровня с огромным адресным пространством.

4.4.7. Фрагментация

В объединяемых сетях разных технологий допустимая максимальная длина пакетов **(Maximum Transfer Unit, MTU)** различна и варьируется от 53 байт в ATM-сетях до 65 535 байт в IP-сетях. При объединении таких сетей возникает проблема, связанная необходимостью разбиения большого пакета при его передаче через сеть с меньшей допустимой длиной пакета. Если пакет проходит через последовательность сетей и попадает в сеть, у которой значение MTU оказывается меньше размера пакета, пограничный маршрутизатор разбивает пакет на две или более части.

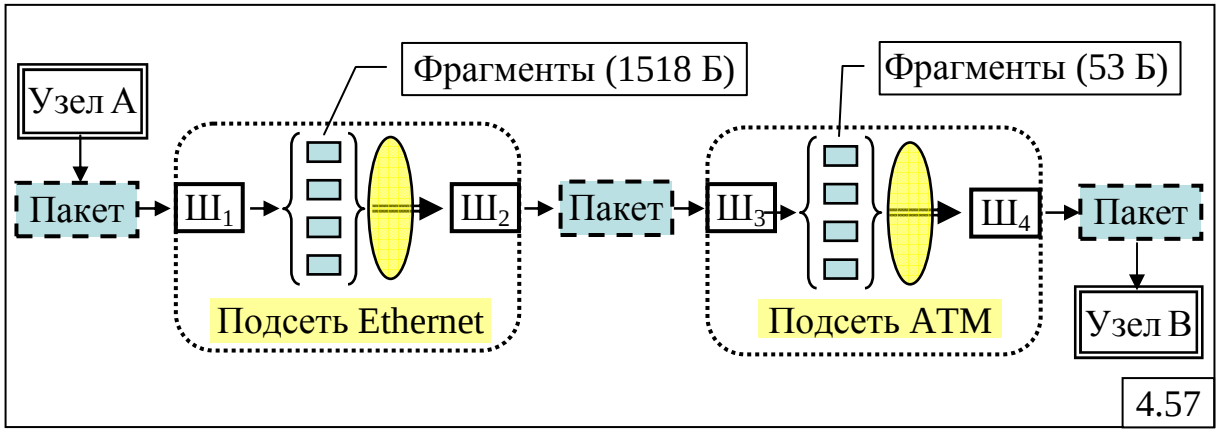
Процесс разбиения длинного пакета на более короткие называется ***фрагментацией***, а соответствующие короткие пакеты – ***фрагментами***. При фрагментации каждый новый пакет получает свой IP-заголовок (20 байт), что увеличивает накладные расходы. После прохождения

фрагментов через соответствующую сеть необходимо восстановить исходный пакет из фрагментов.

Фрагментация в сетях может быть реализована двумя способами (рис.4.56):

- прозрачная фрагментация;
- сквозная фрагментация.

Принцип реализации ***прозрачной фрагментации*** на примере передачи длинного пакета от узла А к узлу В через две подсети (Ethernet и ATM) с меньшим значением MTU показан на рис.4.57.



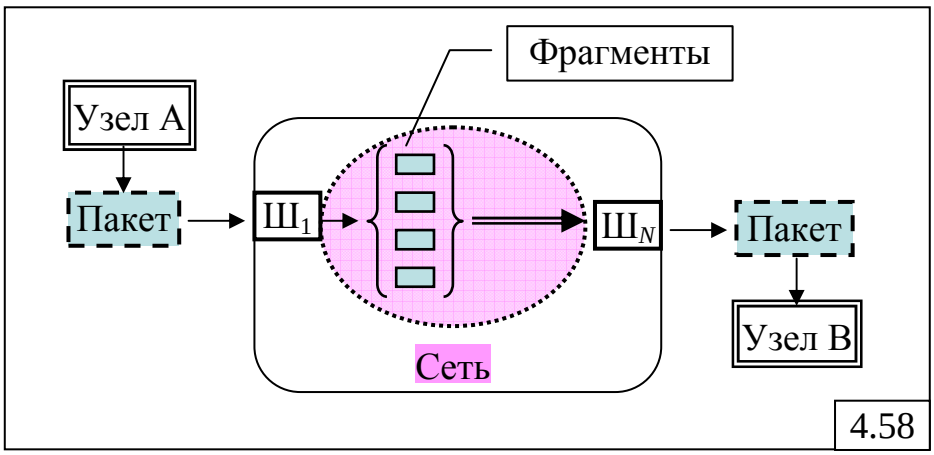
Подсети с разными MTU имеют шлюзы – специализированные маршрутизаторы, предоставляющие интерфейсы для связи с другими сетями. Если на такой шлюз приходит пакет слишком большого размера, он разбивается на фрагменты в соответствии с принятым в данной сети значением MTU. Каждый фрагмент адресуется одному и тому же выходному шлюзу, который восстанавливает из этих фрагментов исходный пакет и. Таким образом, прохождение данных через сети (подсети) с маленькими значениями MTU оказывается прозрачным для пользователей.

Прозрачная фрагментация обладает простотой, но при этом имеет ряд существенных недостатков:

- выходной шлюз должен собрать все фрагменты для восстановления исходного пакета, для чего в заголовках фрагментов необходимо иметь дополнительную информацию, например, номер фрагмента и признак последнего фрагмента;
- все фрагменты одного пакета должны покидать подсеть через один и то же шлюз, что снижает эффективность маршрутизации;

- появляются дополнительные накладные расходы на фрагментацию и дефрагментацию, что снижает производительность сети и увеличивает время доставки пакетов.

Сквозная фрагментация (рис.4.58) является альтернативной по отношению к прозрачной фрагментации и состоит в отказе от восстановления пакета из фрагментов в каждой подсети. Пакет разбивается на фрагменты сразу же в узле-отправителе А или в шлюзе $Ш_1$ сети Эти фрагменты передаются по сети как самостоятельные пакеты независимо друг от друга и собираются только в конечном шлюзе $Ш_N$ или узле-получателе В.



Недостатками такого способа фрагментации являются следующие:

- необходимо, чтобы каждый узел (или шлюз) могли восстанавливать пакеты из фрагментов;
- возрастают накладные расходы на передачу данных, так как каждый фрагмент должен иметь заголовок, который сохраняется на протяжении всего пути, что снижает пропускную способность сети;
- необходимо иметь информацию о том, какие значения MTU имеют подсети, через которые проходит путь передачи данных, чтобы задать размер фрагментов.

Для того чтобы правильно восстановить исходный пакет из фрагментов необходимо иметь эффективную систему нумерации фрагментов. Одна из таких систем основана на понятии ***элементарного фрагмента***, имеющего небольшой размер, достаточный для его передачи через любую подсеть. Например, длина элементарного фрагмента может быть равна 8 байтам, как это показано на рис.4.59. Исходный пакет разбивается на множество элементарных фрагментов одинаковой длины (рис.4.59,а), кроме последнего, который может быть короче. Фрагменты, формируемые в некоторой подсети и называемые ***межсетевыми пакетами***, могут состоять из нескольких элементарных фрагментов (рис.4.59,б), число которых определяется значением MTU, принятым для данной подсети. Заголовок таких фрагментов должен содержать (рис.4.59):

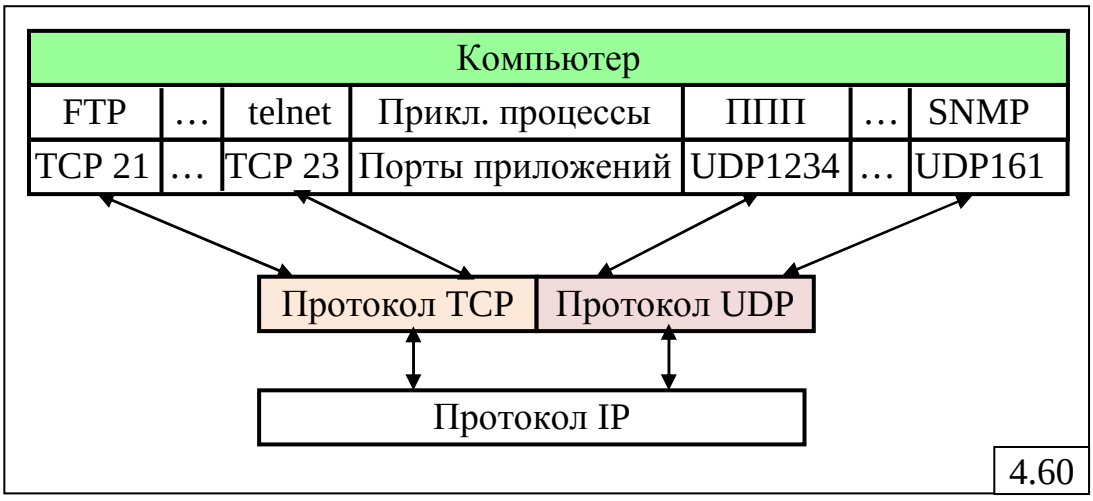
- номер исходного пакета (ИП);

- номер первого элементарного фрагмента (нумерация начинается с нуля), содержащегося в нём, который в заголовке IP-пакета называется *смещением фрагмента* (СФ);
- признак конца (ПК) пакета.

Поскольку размер элементарного фрагмента выбирается таким образом, чтобы он мог пройти через любую сеть, дальнейшая фрагментация межсетевого пакета не составляет проблемы.

4.4.8. Транспортные протоколы стека TCP/IP

Транспортные протоколы TCP и UDP стека протоколов TCP/IP обеспечивают передачу данных между любой парой *прикладных процессов*, выполняющихся в сети, и предоставляют интерфейс для протокола IP путем демультиплексирования нескольких процессов, использующих в качестве адресов транспортного уровня порты. Для каждого прикладного процесса (ПП) (приложения), выполняемого в компьютере, может быть сформировано *несколько точек входа*, выступающих в качестве *транспортных адресов*, называемых ***портами*** (рис.4.60).



Существуют два способа присвоения порта приложению:

- **централизованный** (присвоенные или назначенные номера от 0 до 1023), использующий стандартные номера, присвоенные

общедоступным службам (приложениям), например: FTP – 21, telnet – 23, SMTP – 25, DNS – 53, HTTP – 80.

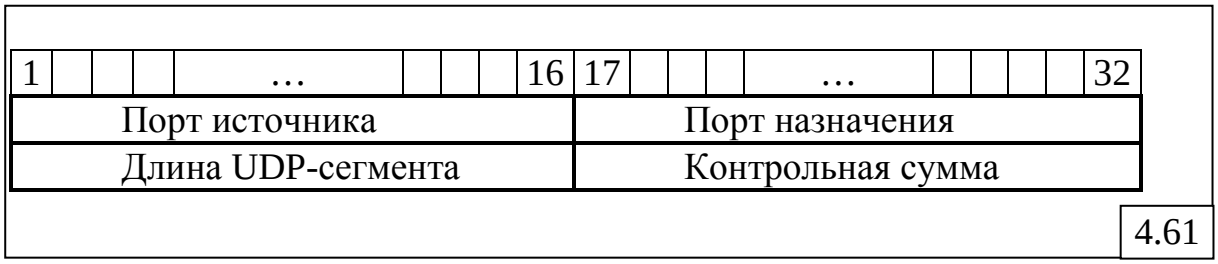
- **локальный** (динамические номера от 1024 до 65535), предоставляющий произвольный номер из списка свободных номеров при поступлении запроса от приложения пользователя.

Динамические номера портов приложений являются уникальными в пределах каждого компьютера, но могут совпадать с номерами портов в других компьютерах. Различие между ними определяется только различием интерфейсов каждого из компьютеров, задаваемых IP-адресами. Таким образом, пара «**IP-адрес; номер порта**», называемая ***сокетом*** (socket), однозначно определяет прикладной процесс в сети.

Номера UDP- и TCP-портов в пределах одного и того же компьютера могут совпадать, хотя и идентифицируют разные приложения. Поэтому при записи номера порта обязательно указывается тип протокола транспортного уровня, например 2345/TCP и 2345/UDP. В некоторых случаях, когда приложение может обращаться по выбору к протоколу UDP или TCP, ему могут быть назначены одинаковые номера UDP- и TCP-портов, например DNS-приложению назначен номер 53 – 53/UDP и 53/TCP.

4.4.8.1. Транспортный протокол UDP

UDP – транспортный протокол, обеспечивающий передачу данных в виде *дейтаграмм* между любой парой *прикладных процессов*, выполняющихся в сети, *без установления соединения*. Сегменты состоят из 8-байтового заголовка, за которым следует поле данных. Заголовок UDP-сегмента показан на рис.4.61.



4.61

Наиболее широко UDP используется при выполнении клиент-серверных приложений (типа запрос-ответ).

При этом UDP не выполняет:

- контроль потока,
- контроль ошибок,
- повторной передачи после получения испорченного сегмента.

Примерами приложений, использующих протокол UDP для передачи данных, являются DHCP, DNS, SNMP.

В некоторых случаях на одном конечном узле может выполняться несколько копий одного и того же приложения. Возникает вопрос: каким образом различаются эти приложения?

Для этого рассмотрим на простом примере процесс формирования запроса и процедуру обращения DNS-клиента к DNS-серверу, когда на одном компьютере запущены два DNS-сервера, причём оба используют для передачи своих данных транспортный протокол UDP (рис.4.62). Для того чтобы различать DNS-серверы, им присваиваются разные IP-адреса – IP1 и IP2, которые вместе с номером порта образуют два разных сокета: «UDP-порт 53, IP1» и «UDP-порт 53, IP2».

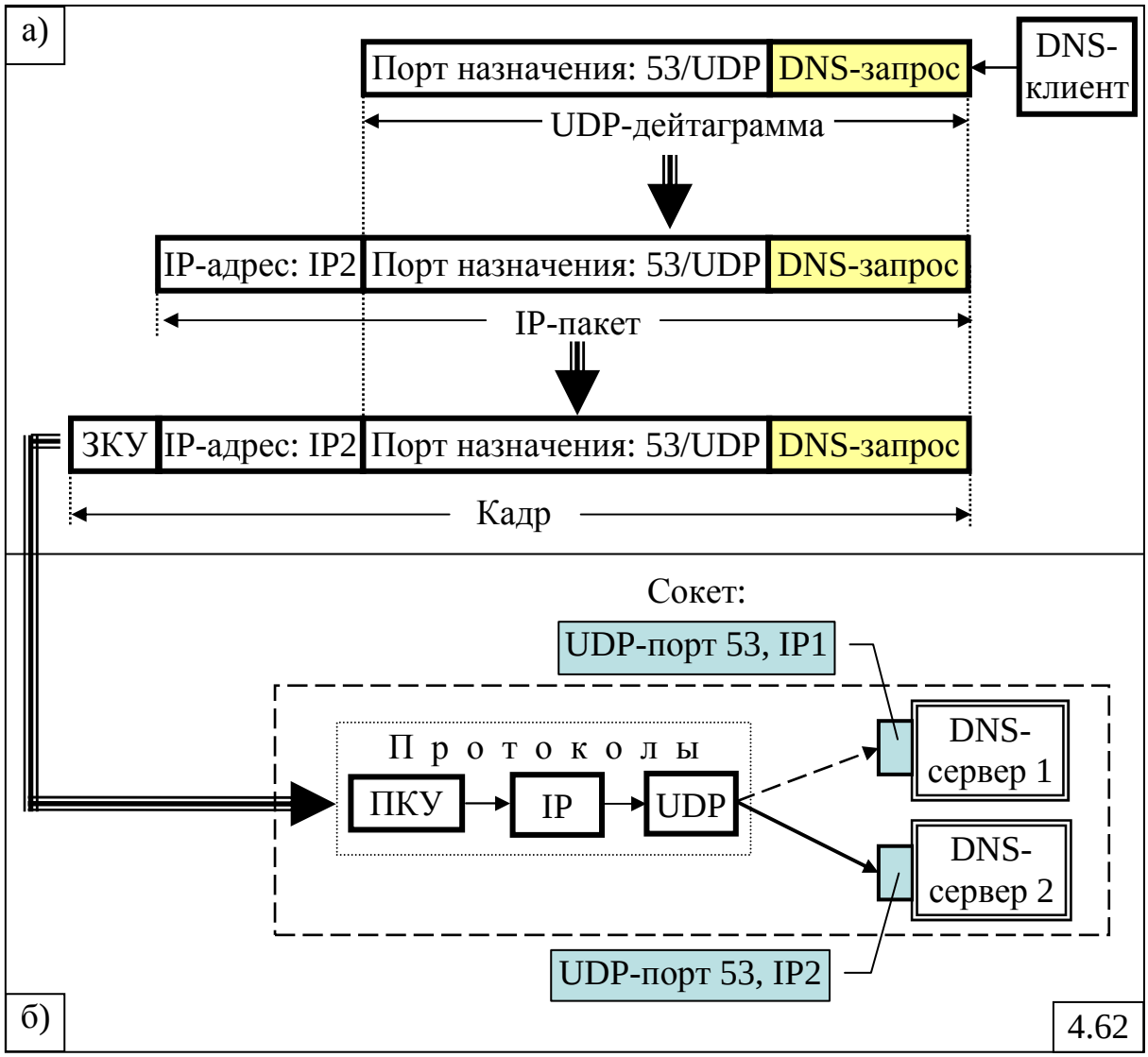


Рис.4.62,а) иллюстрирует процесс формирования DNS-клиентом запроса к DNS-серверу.

DNS-запрос транспортном уровне стека протоколов TCP/IP передаётся протоколу UDP, который вкладывает этот запрос в UDP-дейтаграмму и указывает в заголовке порт назначения 53/UDP. Затем UDP-дейтаграмма передаётся на межсетевой уровень, где она вкладывается в IP-пакет, заголовок которого содержит «IP-адрес: IP2». IP-пакет, в свою очередь, передаётся на уровень «межсетевой интерфейс», где он помещается в кадр канального уровня с соответствующим заголовком канального уровня (ЗКУ). Этот кадр передаётся по сети к компьютеру, содержащему два DNS-сервера (рис.4.62,б).

В этом компьютере протокол канального уровня (ПКУ) снимает заголовок ЗКУ и передаёт содержимое кадра на межсетевой уровень протоколу IP, который, в свою очередь, извлекает содержимое (UDP-дейтаграмму) из IP-пакета. Дальнейшие манипуляции с передаваемыми данными отличаются от принципов, заложенных в многоуровневую модель иерархии протоколов. Вместо того чтобы просто передать UDP-дейтаграмму, находящуюся в поле данных IP-пакета, транспортному уровню, *IP-протокол присоединяет к UDP-дейтаграмме* так называемый **псевдозаголовк**, содержащий среди прочего IP-адреса отправителя и получателя. Таким образом, протокол UDP, имея IP-адрес и порт назначения, однозначно определяет, что содержимое поля данных (то есть DNS-запрос), должно быть передано приложению «DNS-сервер 2».

Назначение и формат псевдозаголовка, который используется также и в TCP-сегменте, описаны в п.4.4.8.3.

4.4.8.2. Транспортный протокол TCP

Протокол TCP обеспечивает надежную передачу данных между прикладными процессами за счет установления логических соединений между взаимодействующими процессами.

Логическое соединение между двумя прикладными процессами идентифицируется парой сокетов (IP-адрес, номер порта), каждый из которых описывает один из взаимодействующих процессов.

Информация, поступающая к протоколу TCP в рамках логического соединения от протоколов более высокого уровня, рассматривается протоколом TCP как *неструктурированный поток байтов* и заносится в буфер. Для передачи на сетевой уровень из буфера вырезается **сегмент**, не превосходящий 64 Кбайт (максимального размера IP-пакета). На практике обычно длина сегмента ограничивается значением 1460 байтами, что позволяет поместить его в кадр Ethernet с заголовками TCP и IP.

Соединение TCP ориентировано на *полнодуплексную передачу*.

Управление потоком данных в протоколе TCP осуществляется с использованием механизма **скользящего окна переменного размера**. При передаче сегмента узел-отправитель включает таймер и ожидает подтверждения. Отрицательные квитанции не посылаются, а используется *механизм тайм-аута*. Узел назначения, получивший сегмент формирует и посылает обратно сегмент (с данными, если они есть, или без данных) с номером подтверждения, равным следующему порядковому *номеру ожидаемого байта*. В отличие от многих других протоколов, протокол TCP подтверждает получение *не пакетов, а байтов* потока. Если время ожидания подтверждения истекает, отправитель посылает сегмент еще раз.

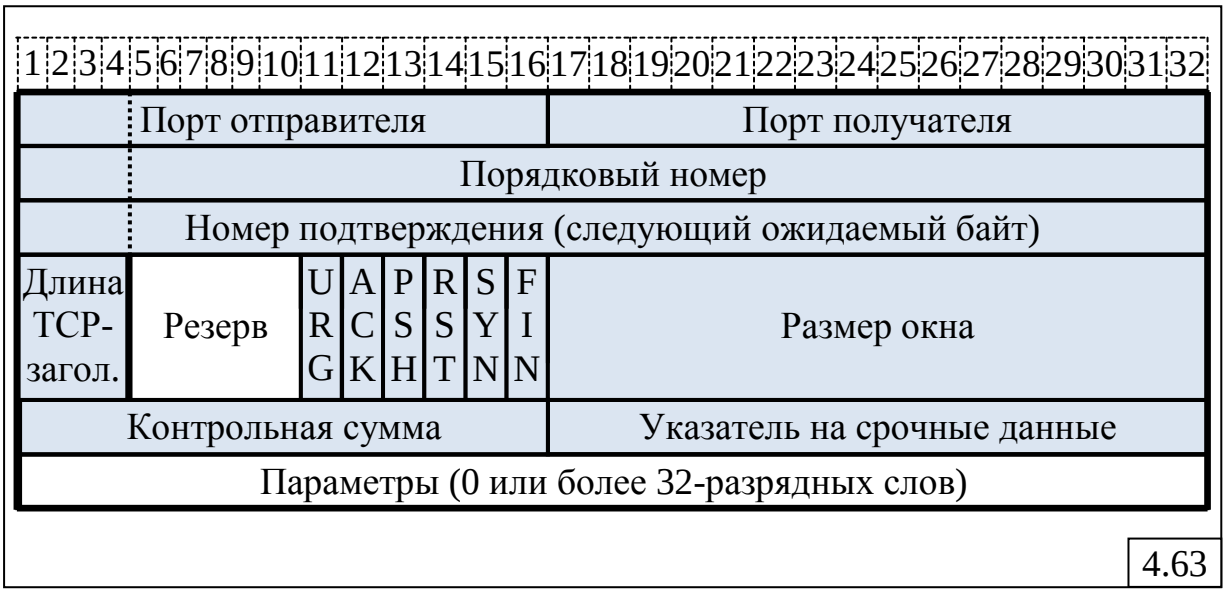
Несмотря на кажущуюся простоту протокола, в нем имеется ряд нюансов, которые могут привести к некоторым проблемам.

Во-первых, поскольку сегменты при передаче по сети могут фрагментироваться, возможна ситуация, при которой часть переданного сегмента будет принята, а остальная часть окажется потерянной.

Во-вторых, сегменты могут прибывать в узел назначения в произвольном порядке, что может привести к ситуации, при которой байты с 2345 по 3456 уже прибыли, но подтверждение для них не может быть выслано, так как байты с 1234 по 2344 еще не получены.

В-третьих, сегменты могут задержаться в сети так долго, что у отправителя истечёт интервал ожидания, и он передаст их снова. Переданный повторно сегмент может пройти по другому маршруту и может быть иначе фрагментирован, или же сегмент может по дороге случайно попасть в перегруженную сеть. В результате для восстановления исходного сегмента потребуется достаточно сложная обработка

На рис.4.63 представлен формат заголовка TCP-сегмента. Первые 20-байт заголовка имеют строго фиксированный формат, за которым могут находиться дополнительные поля. После дополнительных полей заголовка размещается поле данных, содержащее не более 65 495 байт, которое вместе с TCP- и IP-заголовками размером по 20 байт даст максимально допустимый размер IP-пакета в 65 535 байт.



4.63

Не вдаваясь в детали, рассмотрим кратко назначение фиксированных полей заголовка TCP-сегмента.

Поля «Порт отправителя» (2 байта) и «Порт получателя» (2 байта) идентифицируют *процессы*, между которыми установлено логическое соединение.

Поле «Порядковый номер» (4 байта) содержит *номер первого байта* данных в сегменте, который определяет смещение сегмента относительно потока передаваемых данных

Поле «Номер подтверждения» (4 байта) содержит *номер следующего ожидаемого байта*, который используется в качестве квитанции, подтверждающей правильный приёма всех предыдущих байтов.

Поле «Длина TCP-заголовка» (4 бита) задаёт длину заголовка TCP-сегмента, измеренную в 32-битовых словах.

Поле «Резерв» длиной 6 бит зарезервировано на будущее.

Однобитовые **флаги** несут служебную информацию о типе сегмента и интерпретируются следующим образом:

URG=1 указывает на наличие *срочных данных*, что означает использование поля «Указатель на срочные данные»;

ACK=1 означает, что сегмент является *квитанцией* на принятый сегмент и поле «Номер подтверждения» содержит осмысленные данные. В противном случае данный сегмент не содержит подтверждения и поле «Номер подтверждения» просто игнорируется.

PSH=1 (PUSH-флаг) означает *запрос на отправку данных* без ожидания заполнения буфера;

RST=1 используется для *сброса состояния соединения* при обнаружении проблем, а также для отказа от неверного сегмента или от попытки создать соединение;

SYN=1 используется для *установки соединения*, при этом если ACK=0, то это означает, что поле подтверждения не используется;

FIN=1 используется для *разрыва соединения*.

Поле «Размер окна» (2 байта) определяет, сколько байт может быть послано после байта, получившего подтверждение.

Поле «Контрольная сумма» (2 байта) содержит контрольную сумму, которая охватывает заголовок, данные и *псевдозаголовок*.

Алгоритм вычисления контрольной суммы выглядит следующим образом.

Перед началом вычисления контрольной суммы значение этого поля устанавливается равным нулю. Если поле данных содержит нечётное число байтов, то оно дополняется нулевым байтом, который используется при подсчёте контрольной суммы, но не вставляется в сегмент для передачи в сети. Необходимость такого добавления обусловлена тем, что TCP-сегмент, включающий заголовок, данные и псевдозаголовок, рассматривается как совокупность 16-разрядных двоичных чисел, которые складываются в дополнительном коде, а затем вычисляется дополнение для полученной суммы, которое заносится в поле «Контрольная сумма». Получатель сегмента аналогичным образом подсчитывает контрольную сумму для всего сегмента, включая поле «Контрольная сумма». Очевидно, что полученный таким образом результат должен быть равен 0. Отметим, что дополнительный нулевой байт

Поле «Указатель на срочные данные» (2 байта) содержит смещение в байтах от текущего порядкового номера байта до места расположения срочных данных, которые необходимо срочно принять, несмотря на переполнение буфера. Таким образом, в протоколе TCP реализуются прерывающие сообщения. Содержимым срочных данных занимается прикладной уровень. Протокол TCP лишь обеспечивает их доставку и не интересуется причиной прерывания.

Поле «Параметры» имеет переменную длину и может отсутствовать.

Примерами приложений, использующих протокол TCP для передачи данных, являются FTP, TFTP, DNS, POP3, IMAP, TELNET.

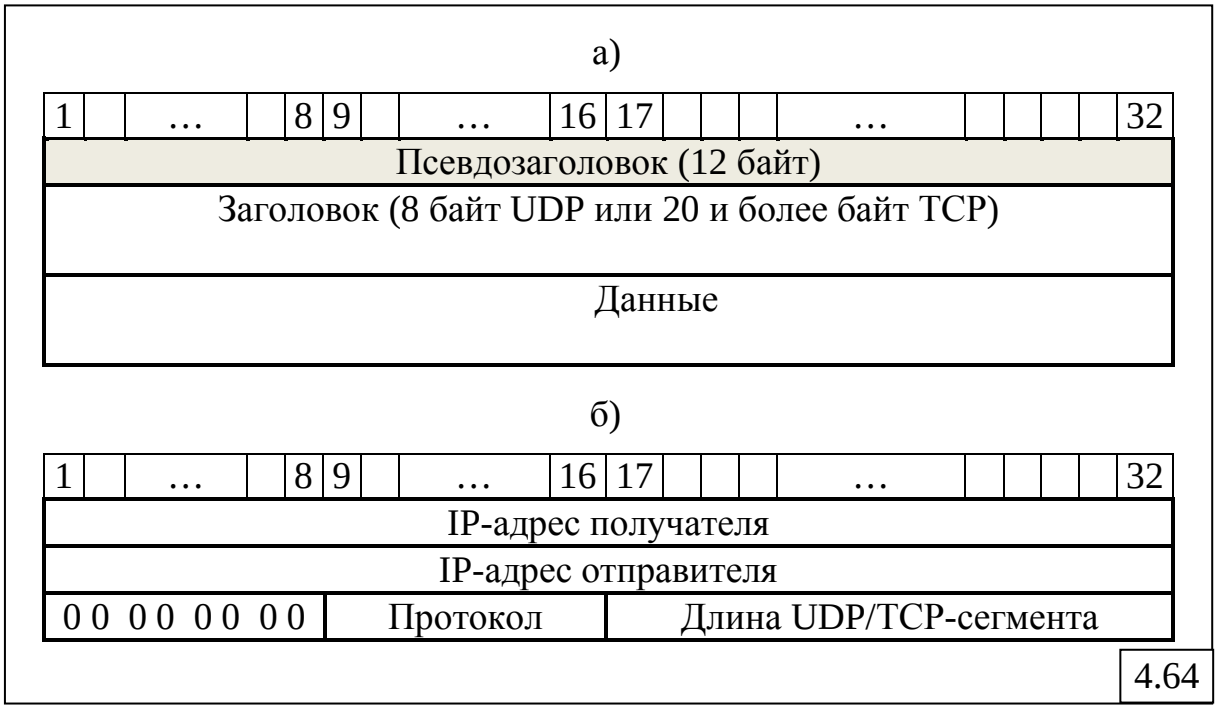
4.4.8.3. Псевдозаголовок протоколов UDP и TCP

Как сказано выше, при передаче данных от нижележащего межсетевого уровня на транспортный уровень в заголовки UDP-дейтаграмм и TCP-сегментов включается псевдозаголовок, который располагается перед основным заголовком транспортного уровня. Таким образом, блок данных транспортного уровня (UDP-дейтаграмма или TCP-сегмент) будет содержать (рис.4.64,а):

- псевдозаголовок длиной 12 байт или 3 32-разрядных слова;
- заголовок длиной 8 байт для UDP-дейтаграммы или 20 и более байт для TCP-сегмента;
- данные.

Псевдозаголовок, формат которого показан на рис.4.64,б, содержит:

- IP-адрес отправителя;
- IP-адрес получателя;
- нулевое поле, не используемое и заполненное нулями;
- поле «Протокол», содержащее номер протокола транспортного уровня: 17 для протокола UDP и 6 для протокола TCP;
- длина UDP-дейтаграммы или TCP-сегмента.



Включение псевдозаголовка в контрольную сумму блока данных транспортного протокола помогает обнаружить неверно доставленные пакеты за счёт двойной проверки, выполняемой протоколом IP и протоколами транспортного уровня. Кроме того, передача IP-адресов транспортному уровню позволяет однозначно разрешить ситуацию, показанную на рис.4.62, когда две копии одного и того же приложения используют одинаковый номер порта.

Узел-отправитель при формировании TCP-сегмента рассчитывает контрольную сумму сегмента с учётом псевдозаголовка. Однако при

передаче по сети *псевдозаголовок не включается в сегмент*, что позволяет уменьшить накладные расходы и, соответственно, повысить эффективную скорость передачи пользовательских данных. В узле-получателе протокол IP формирует псевдозаголовок и вставляет его в поступивший сегмент и передаёт транспортному уровню.

4.4.9. Управляющий протокол ICMP

Internet Control Message Protocol (ICMP) – **протокол межсетевых управляющих сообщений** предназначен для выявления и обработки нештатных событий (например, потеря пакета), заключающейся в определении типа ошибки, формировании сообщения о ней и передаче этого сообщения приложению, сформировавшему пакет.

К основным функциям протокола ICMP относятся:

- обмен тестовыми сообщениями для выяснения наличия и активности узлов сети;
- анализ достижимости узла-получателя и сброс пакетов, направляемых к недостижимым узлам;
- изменение маршрутов;
- уничтожение пакетов с истекшим временем жизни;
- синхронизация времени в узлах сети;
- управление потоком путем регулирования частоты посылки пакетов узлами-источниками.

Основные типы ICMP-сообщений:

- «адресат недоступен» – пакет не может быть доставлен;
- «время истекло» – время жизни пакета достигло нуля;
- «проблема с параметром» – ошибка в поле заголовка;
- «переадресовать» – научить маршрутизатор;
- «запрос отклика» – запрос: жив ли компьютер?;
- «отклик» – да, жив.

Одной из наиболее интересных среди перечисленных функций является изменение маршрутов: если некоторый маршрутизатор определяет, что хост использует неоптимальный путь для доставки пакета, он при помощи протокола ICMP может скорректировать маршрутную таблицу хоста. Это один из механизмов автоматической оптимизации и адаптации сетей TCP/IP к изменениям топологии.

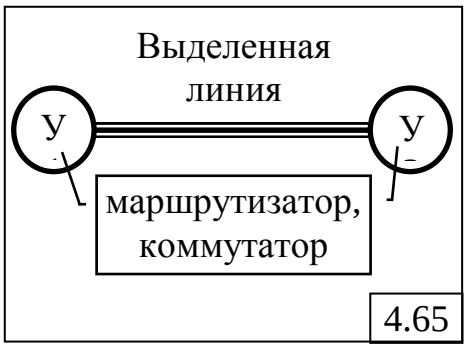
ICMP-пакеты инкапсулируются в IP пакеты. ICMP является неотъемлемой частью IP, но при этом не делает протокол IP средством надёжной доставки сообщений. Для этих целей существует протокол TCP.

4.4.10. Протоколы канального уровня для выделенных линий

Протоколы канального уровня для выделенных линий (рис.4.65) должны:

- обеспечивать надежную передачу;

- предоставлять возможность управления потоком кадров для предотвращения переполнения соседних узлов.



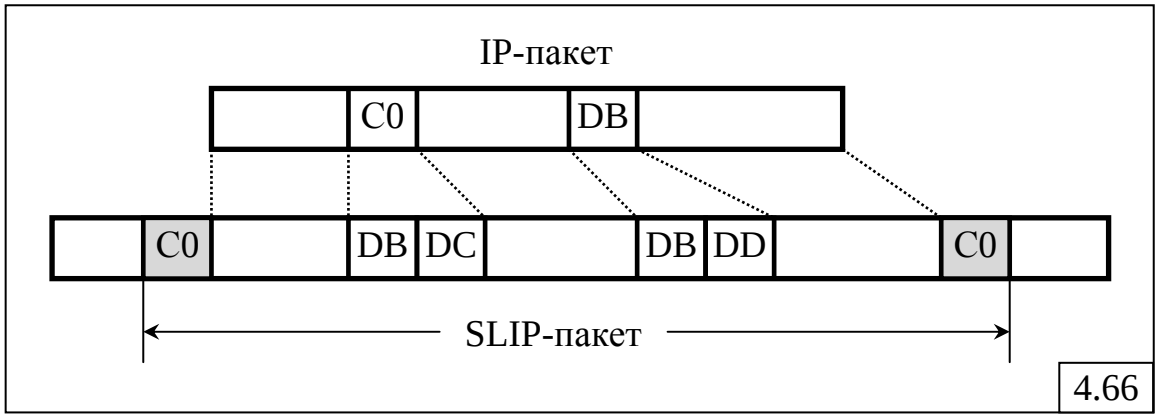
Протоколы канального уровня:

- SLIP;
- протоколы семейства HDLC;
- PPP.

4.4.10.1. Протокол SLIP

SLIP (Serial Line IP) – первый стандарт для протоколов TCP/IP, который может использоваться как для коммутируемых, так и для выделенных каналов ввиду простоты.

SLIP поддерживается только протоколом сетевого уровня IP. Основная и единственная *функция* протокола SLIP – распознавание начала и конца IP-пакета в потоке бит. Для этого в качестве границ IP-пакета используется специальный символ END (шестнадцатеричный код – $C0_{16}$). Если в IP-пакете встречается код $C0_{16}$, то используется процедура ***байт-стаффинга*** (рис.4.66), заключающаяся в следующем. Код $C0_{16}$ заменяется на коды DB_{16} и DC_{16}, а код DB_{16} заменяется на DB_{16} и DD_{16}.



К недостаткам протокола SLIP относятся:

- отсутствие возможности обмениваться адресной информацией;
- использование только IP-пакетов в качестве содержимого SLIP-пакета;
- отсутствие процедур обнаружения и коррекции ошибок.

4.4.10.2. Протоколы семейства HDLC

HDLC (High-level Data Link Control Procedure) – высокоуровневый протокол управления каналом – стандарт ISO для выделенных линий. Представляет собой *семейство протоколов LAP* (Link Access Protocol), включающее следующие протоколы:

- **LAP-B** – для сетей X.25 (B – Balanced);
- **LAP-D** – для сетей ISDN (D – D-channel);
- **LAP-M** – для модемов (M – Modem);

- **LAP-F** – для сетей Frame Relay (F – Frame Relay).

HDLC относится к бит-ориентированным протоколам и использует кадр, формат которого показан на рис.4.67.

0111111	**Адрес**	**У**	**Данные**	**КС**	0111111	
8	8/16	8/16	-	16	8	бит

4.67

В качестве обрамления кадра, служащих границами между передаваемыми кадрами, используется специальная последовательность из 8 бит (байт): 01111110, называемая ***флагом***. Благодаря наличию флагов нет необходимости указывать длину кадра. Для того, чтобы отличать последовательность бит 01111110, находящуюся в поле данных от флага применяется процедура *бит-стаффинга*.

Поле **Адрес** имеет длину 1 или 2 байта и при наличии нескольких узлов-приёмников используется для идентификации конкретного узла, а в двухточечном соединении – для того, чтобы отличить команды от ответов, а также для указания направления передачи кадра по интерфейсу «пользователь – сеть».

Поле **Данные** может быть любой длины и содержать пакеты протоколов вышележащих уровней. Это поле может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.

Поле **КС** (контрольная сумма) содержит *остаток избыточной циклической суммы*, вычисленной с помощью полиномов типа CRC.

Поле **У** (*управление*) имеет длину 1 или 2 байта и содержит служебную информацию. Структура и содержимое этого поля зависят от типа передаваемого HDLC-кадра.

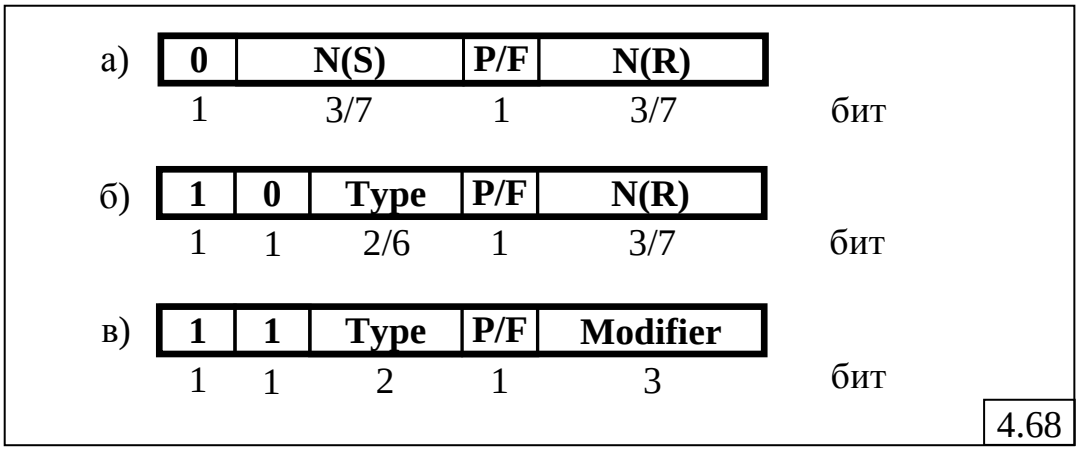
Существуют 3 типа HDLC-кадров, различающиеся содержимым поля **У** (*управление*), показанного на рис.4.68:

- *информационные кадры* длиной 1 или 2 байта (рис.4.68,а), предназначенные для передачи данных пользователя;
- *управляющие* или *супервизорные кадры* длиной 1 или 2 байта (рис.4.68,б), предназначенные для передачи команд и ответов в процессе установленного логического соединения;
- *ненумерованные кадры* длиной 1 байт (рис.4.68,в), предназначенные для установления и разрыва логического соединения, а также информирования об ошибках.

Тип кадра определяется первыми битами поля управления: 0 – информационный кадр; 10 – управляющий кадр; 11 – ненумерованный кадр.

Протокол HDLC для обеспечения надёжности передачи данных использует механизм скользящего окна, ширина которого составляет:

- 7 кадров при длине поля управления в 1 байт;
- 127 кадров при длине поля управления в 2 байта.



4.68

Для реализации механизма скользящего окна в **информационном кадре** предусмотрено 2 поля:

- поле **N(S),** содержащее порядковый номер передаваемого кадра;
- поле **N(R),** содержащее номер очередного ожидаемого кадра.

Наличие этих двух полей связано с реализацией дуплексной передачи данных, а их длина определяет ширину окна в 7 (при длине 3 бита) или 127 (при длине 7 бит) кадров.

Бит **P/F** (Poll/Final – Опрос/Финал) используется для указания промежуточного (P) или последнего передаваемого (F) кадра. В некоторых случаях этот бит может использоваться для указания другому узлу о необходимости передать управляющий кадр, не ожидая попутного потока данных.

Управляющие кадры могут быть 4-х типов, которые различаются значением поля **Type:**

- **Type=0** – *подтверждение* (RESEIVE READY – к приёму готов) – передаёт в поле **N(R)** номер следующего ожидаемого кадра и используется при отсутствии попутного потока данных для передачи подтверждения;
- **Type=1** – *отрицательное подтверждение* (REJECT – отказ) – передаёт в поле **N(R)** номер неверно полученного кадра, начиная с которого узел-отправитель должен повторить передачу кадров;
- **Type=2** – *отказ* (RESEIVE NOT READY – к приёму не готов) – означает, что в узле-получателе возникли проблемы, не позволяющие принимать кадры (например, переполнена буферная память) и, соответственно, узел-отправитель должен приостановить передачу кадров, при этом в поле **N(R)** указывается номер кадра, начиная с которого узел-отправитель в дальнейшем (после устранения причины приостановки приёма кадров) должен будет повторить передачу кадров;
- **Type=3** – *выборочное подтверждение* (SELECTIVE REJECT – выборочный отказ) – передаёт в поле **N(R)** номер только того кадра, передачу которого узел-отправитель должен повторить.

Ненумерованные кадры применяются в основном для служебных целей, но могут переносить и данные, когда требуется ненадёжный не требующий соединения сервис. Поля **Type** и **Modifier** определяют типы и

модификации команд, используемых двумя узлами на этапе установления соединения. Примерами таких команд могут служить:

- запрос на установление соединения с использованием двухбайтовых полей управления для информационных и управляющих кадров: SABME (Set Asynchronous Balanced Mode Extended – установить асинхронный сбалансированный расширенный режим);
- подтверждение установления или разрыва соединения: UA (Unnumbered Acknowledgment – ненумерованная положительная квитанция);
- запрос на разрыв соединения: REST (Resetting connection – сброс соединения).

Одна из основных функций протоколов семейства HDLC – восстановление искаженных и потерянных кадров (уменьшение вероятности искажения бита – BER с 10^{-3} – 10^{-4} до 10^{-9}). Для современных каналов высокого качества, обеспечивающих значение BER=10^{-8} – 10^{-9}, использование протоколов семейства HDLC на уровне моста или маршрутизатора становится нецелесообразным.

4.4.10.3. Протокол PPP

PPP (Point-to-Point Protocol) – протокол двухточечного соединения, заменивший протокол SLIP и построенный на основе формата кадров протоколов семейства HDLC с дополнением собственных полей.

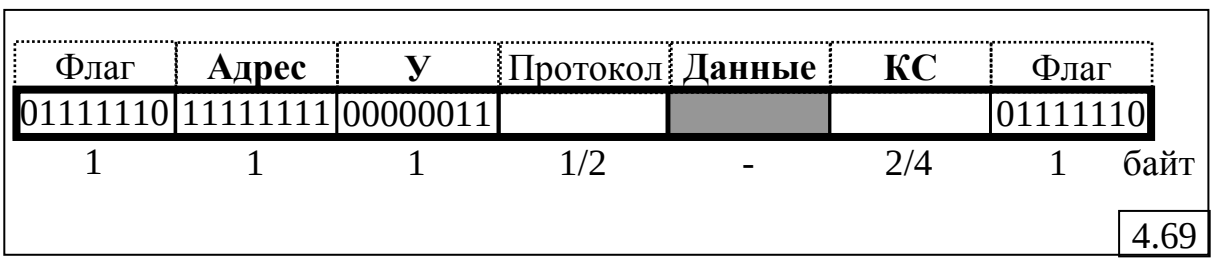
Протокол PPP является стандартным протоколом Интернета и так же, как протокол HDLC, представляет собой семейство протоколов, включающее в том числе:

- LCP (Link Control Protokol) – протокол управления соединением;
- NCP (Network Control Protokol) – протокол управления сетью;
- MLPPP (Multi Link PPP) – многоканальный протокол PPP.

Протокол PPP основан на четырех принципах:

- переговорное принятие параметров соединения;
- многопротокольная поддержка;
- расширяемость протокола;
- независимость от глобальных служб.

В отличие от бит-ориентированного протокола HDLC, протокол PPP является *байт-ориентированным*, что означает посимвольное заполнение кадра, то есть все кадры состоят из целого числа байтов. Полный формат кадра PPP для работы в ненумерованном режиме показан на рис.4.69.



4.69

Характерными для PPP-кадра являются следующие особенности.

1. Если в поле **Данные** встречается байт 01111110, совпадающий с кодом флага, то используется процедура *байт-стаффинга*, рассмотренная выше.

2. Поле **Адрес** всегда содержит значение 11111111, что означает, что все станции должны принимать этот кадр и позволяет избежать необходимости назначения адресов для передачи данных.

3. Поле **управления У** по умолчанию содержит значение 00000011, означающее ненумерованный кадр.

4. Поле **Протокол** содержит код протокола вышележащего уровня, пакет которого вложен в поле данных. Длина этого поля по умолчанию составляет 2 байта, но путём переговоров длина может быть уменьшена до 1 байта.

5. Поле **Данные** может быть переменной длины, вплоть до некоторого установленного пользователями максимального значения, которое по умолчанию обычно составляет 1500 байт.

6. Поле **контрольной суммы КС** по умолчанию имеет длину 2 байта, которая в случае необходимости по договорённости может быть увеличена до 4-х байтов.

7. Установление соединения между двумя узлами сопровождается сложной переговорной процедурой принятия параметров соединения, таких как качество линии связи, размер передаваемых кадров, тип протокола аутентификации и т.д. Эта переговорная процедура реализуется протоколом управления линией связи LCP.

8. Протокол PPP реализует многопротокольную поддержку, обеспечивая внутри одного соединения передачу пакетов различных протоколов сетевого (IP. IPX, XNS и т.д.) и канального уровня ЛВС.

4.5. MPLS-технология

4.5.1. Основные принципы MPLS-технологии

MPLS – MultProtocol Label Switching – многопротокольная коммутация на основе меток объединяет два способа передачи пакетов: дейтаграммный и «виртуальный канал».

В основе MPLS-технологии лежит технология IP-коммутации (IP-Switching), предложенная в середине 90-х годов компанией IPSILON, которая для её реализации разработала специальное комбинированное устройство IP/ATM, представляющее собой ATM-коммутаторы со встроенными блоками IP-маршрутизации. Эти устройства были предназначены для уменьшения задержек при передаче кратковременных потоков данных за счёт отказа от предварительной процедуры установления виртуального канала, как это происходит в ATM-сетях. Для этого IP-пакет, принадлежащий кратковременному потоку, разбивался устройством IP/ATM на ATM-ячейки, которые передавались от одного устройства IP/ATM к другому. В то же время, долговременные потоки

передавались традиционным для ATM-технологии способом – с предварительным установлением виртуального канала.

Дальнейшие усовершенствования IP-коммутации привели в конце 90-годов прошлого века к созданию технологии MPLS, объединяющей достоинства техники виртуальных каналов и стека протоколов TCP/IP за счёт применения специального сетевого устройства - ***коммутирующего по меткам маршрутизатора*** **LSR (Label Switch Router)**, выполняющего функции *IP-маршрутизатора* и *коммутатора виртуальных каналов*.

4.5.2. Маршрутизатор LSR и таблица продвижения

В основе MPLS лежит принцип передачи на основе меток. Любой передаваемый пакет ассоциируется с тем или иным *классом сетевого уровня* (Forwarding Equivalence Class, FEC), каждый из которых идентифицируется определенной меткой. Значение метки уникально лишь для участка пути между соседними узлами сети MPLS, которыми являются **LSR**. Метка передается в составе любого пакета, причем способ ее привязки к пакету зависит от используемой технологии канального уровня.

LSR получает топологическую информацию о сети, участвуя в работе алгоритма маршрутизации (OSPF, BGP, IS-IS). Затем он начинает взаимодействовать с соседними LSR, распределяя метки, которые в дальнейшем будут применяться для коммутации. Обмен метками может производиться с помощью как специального ***протокола распределения меток*** **LDP (Label Distribution Protocol)**, так и модифицированных версий протоколов сигнализации в сети (например, видоизмененных протоколов маршрутизации, резервирования ресурсов RSVP и др.).

Распределение меток между LSR приводит к установлению внутри домена MPLS ***путей с коммутацией по меткам*** **LSP (Label Switching Path)**, которые хранятся в каждом маршрутизаторе LSR в виде таблицы продвижения (рис.4.70), содержащей следующие столбцы:

- **входной интерфейс** – интерфейс (порт), по которому пакет поступил в LSR;
- **метка** – идентификатор (метка), который идентифицирует принадлежность поступившего пакета к конкретному трафику;
- **следующий хоп** – интерфейс (порт), в который должен быть направлен пакет;
- **действие** – указатель, определяющий, какое действие должно быть применено к метке (заменить, удалить).

Вх. интерфейс	Метка	След.хоп	Действие
I1	121	I2	211
I2	164	I3	274

4.70

В поле «Действие» таблицы продвижения указываются основные *операции с метками*:

- Push – поместить метку в стек;
- Swap – заменить текущую метку новой;
- Pop – удаление верхней метки.

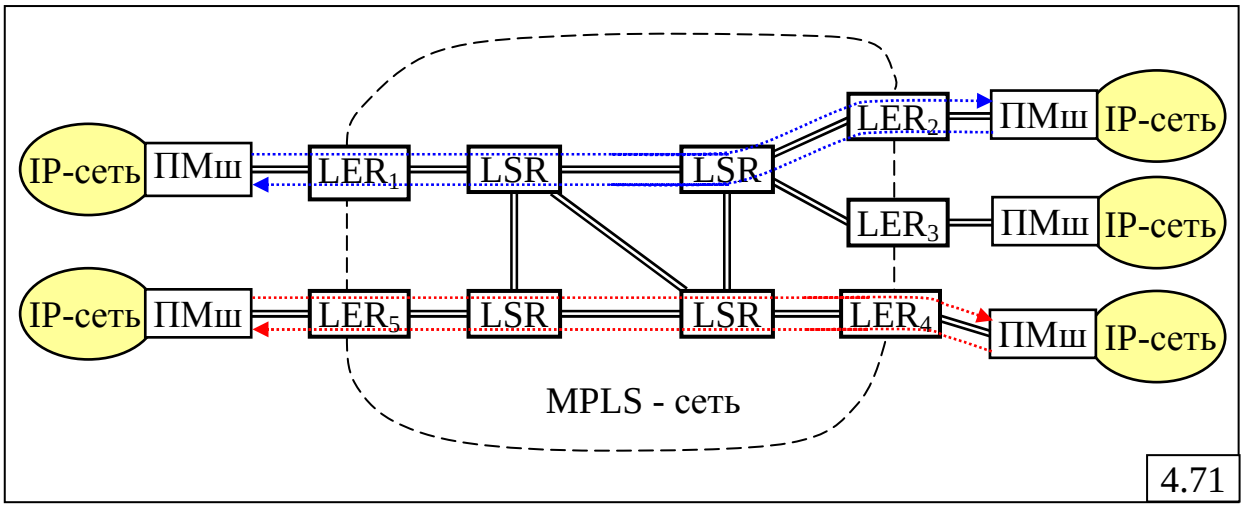
Получая пакет, LSR по номеру интерфейса, на который пришел пакет, и по значению привязанной к пакету метки определяет для него выходной интерфейс. Старое значение метки заменяется новым, содержавшимся в поле «действие» таблицы, и пакет отправляется к следующему устройству на пути LSP.

Вся операция требует лишь одноразовой идентификации значений полей в одной строке таблицы. Это занимает гораздо меньше времени, чем сравнение IP-адреса отправителя с наиболее длинным адресным префиксом в таблице маршрутизации, которое используется при традиционной маршрутизации.

На рис.4.71 показан пример MPLS-сети, находящейся в окружении IP-сетей. Каждая IP-сеть соединяется через *пограничный маршрутизатор* (ПМш) с ***пограничным коммутирующим по меткам маршрутизатором*** **LER (Label switch Edge Routers)**, который выполняет следующие функции:

- классификация пакетов по различным *классам эквивалентного продвижения* (FEC – Forwarding Equivalence Class), имеющих один и тот же следующий хоп;
- реализация таких дополнительных сервисов, как фильтрация, явная маршрутизация, выравнивание нагрузки и управление трафиком.

В результате интенсивные вычисления приходятся на граничную область MPLS-сети, а высокопроизводительная коммутация выполняется в ядре, содержащем множество LSR.

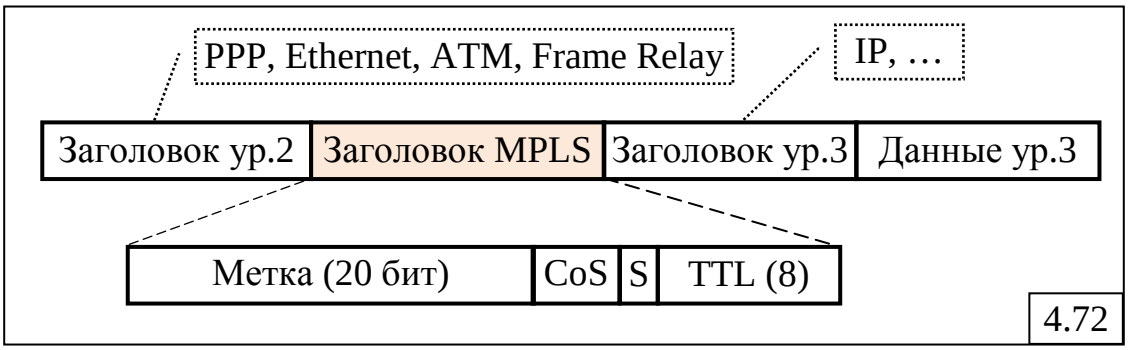


4.5.3. Заголовок MPLS

Заголовок MPLS длиной 32 бита вставляется между заголовками 2-го и 3-го уровня OSI-модели, что даёт повод говорить, что MPLS – это технология уровня 2,5.

Заголовок MPLS содержит (рис.4.72) следующие поля:

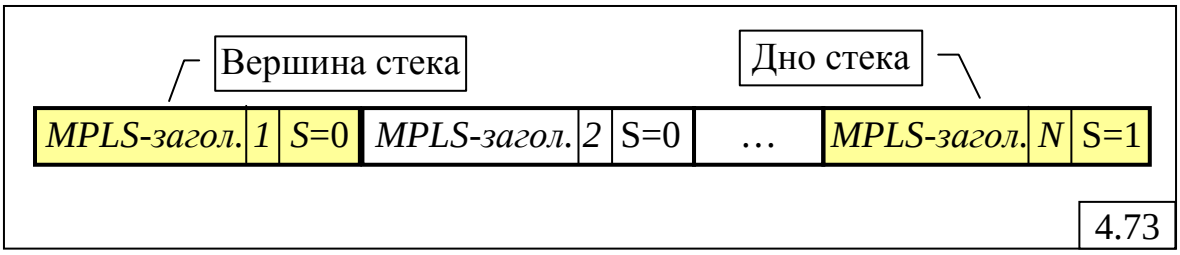
- **метка** (20 бит), на основе которой осуществляется коммутация пакетов в MPLS-сети;
- **CoS** (Class of Service) – класс обслуживания (3 бита), указывающий класс трафика, требующего определённого показателя QoS;
- **S** – признак дна стека меток (1 бит), используемый для организации агрегированных путей LSP при прохождении пакетом через несколько MPLS-сетей;
- **TTL** (Time To Live) – время жизни пакета (8 бит), дублирующее аналогичное поле IP-пакета, что позволяет маршрутизаторам LSR отбрасывать пакеты с истекшим временем жизни.



4.5.4. Многоуровневая коммутация по меткам

Для создания системы агрегированных путей LSP с любым количеством уровней иерархии заголовок MPLS-кадра формируется в виде ***стека меток***, включающего столько заголовков MPLS, сколько уровней иерархии содержит агрегированный путь (рис.4.73). При этом различают:

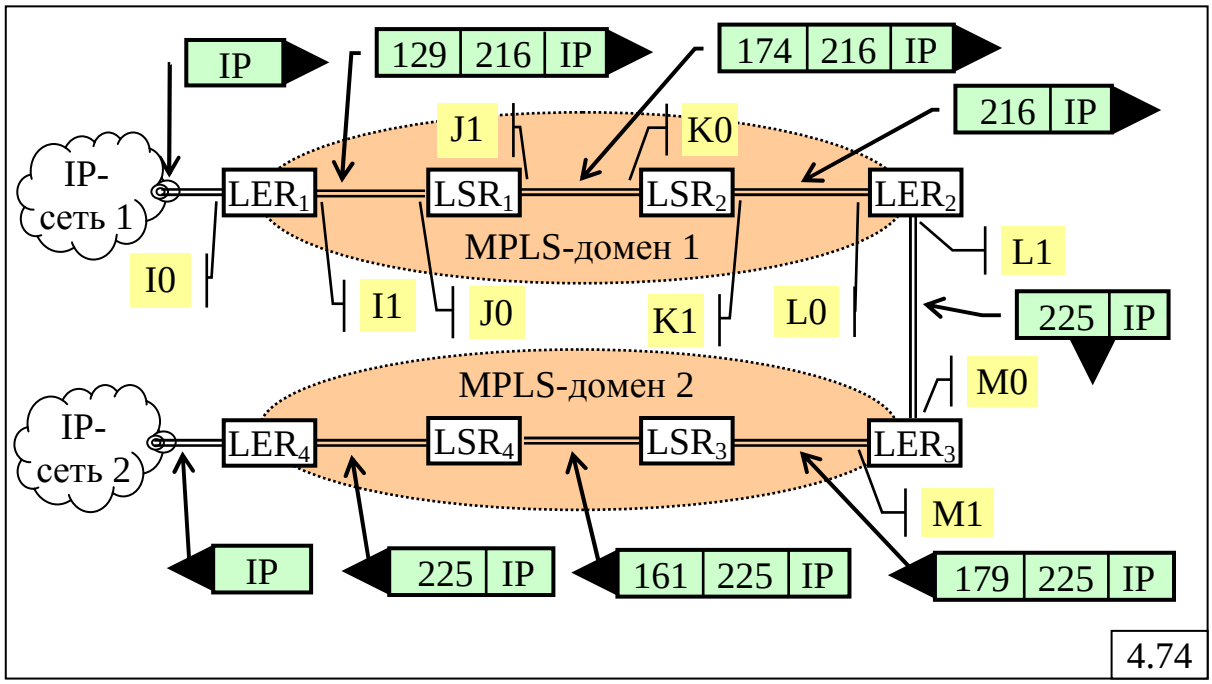
- **вершину стека**, над которым всегда выполняются действия, указанные в таблице продвижения;
- **дно стека**, признаком которого служит значение поля S=1.



Организация стека меток необходима для организации ***многоуровневой коммутации по меткам***, когда пакеты передаются не только внутри каждого MPLS-домена, но и между разными MPLS-доменами, обслуживаемых разными поставщиками услуг. С помощью стека меток может быть реализован механизм *туннелирования*.

Рассмотрим механизм формирования многоуровневой коммутации с использованием стека меток на примере сети, показанной на рис.4.74.

Положим, что на пути передачи **IP-пакета** из **IP-сети 1** в **IP-сеть2** имеются 2 **MPLS-домена**, в каждом из которых пакет проходит через 2 пограничных маршрутизатора LER и 2 маршрутизатора LSR.



4.74

Соответствующие фрагменты таблиц продвижения пакетов маршрутизаторов LER1, LSR1, LSR2 и LER2, для наглядности объединённые в одну таблицу, представлены на рис.4.75.

Маршрутизатор	Входной интерфейс	Метка	Следующий хоп	Действия
LER1	…			
	I0	-	I1	216 | Push | 129
	…			
LSR1	…			
	J0	129	J1	Swap 174
	…			
LSR2	…			
	K0	174	K1	Pop
	…			
LER2	…			
	L0	216	L1	Swap 225
	…			

4.75

IP-пакет из IP-сети 1 по интерфейсу I0 попадает в пограничный маршрутизатор LER1, где в заголовок IP-пакета будет вставлен MPLS-заголовок. В соответствии с таблицей продвижения маршрутизатора LER1 будет сформирован MPLS-заголовок, в поле метки которого будет установлено значение 216. Затем действие Push приведёт к формированию второго MPLS-заголовка, который станет вершиной стека, в поле метки которого будет установлено значение 129. Таким образом, появится стек из двух MPLS-заголовков (см. рис.4.74), причём во втором заголовке

признак дна стека S будет установлен в 1. Далее этот пакет направляется на интерфейс I1, через который он попадёт в маршрутизатор LSR1.

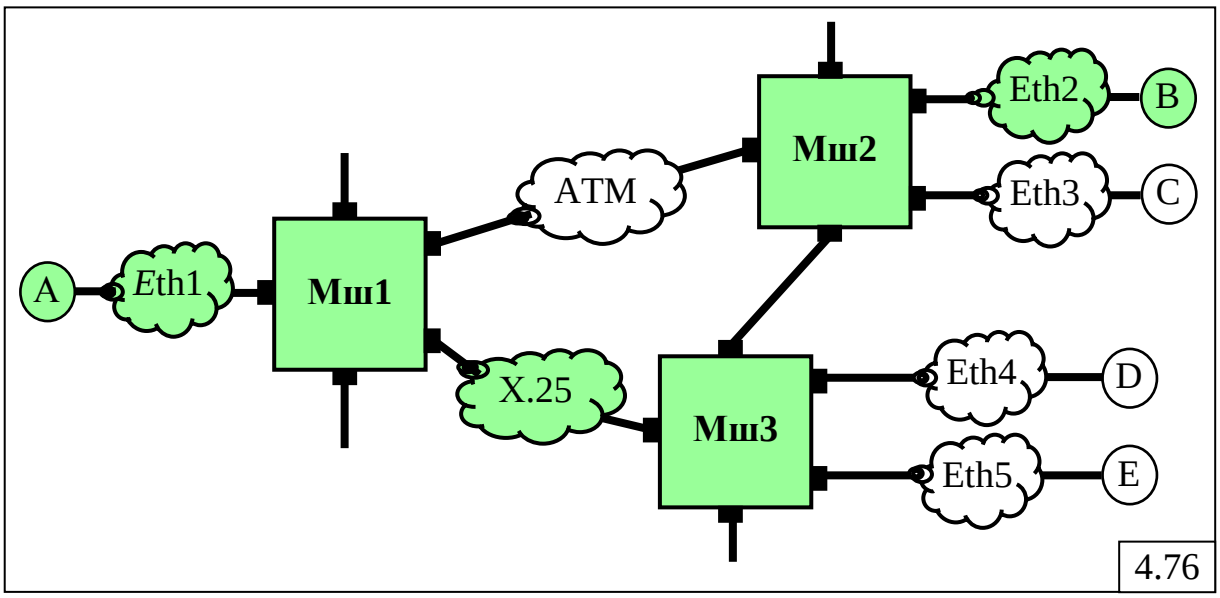
В соответствии с таблицей продвижения LSR1 поступивший по интерфейсу J0 пакет с меткой 129 должен быть направлен в выходной интерфейс J1, при этом метка 129, находящаяся в вершине стека, должна быть заменена на 174 (действие Swap 174).

В маршрутизаторе LSR2 будет удалена (действие Pop) верхняя метка 174, а в пограничном маршрутизаторе LER2 метка 216 будет заменена на 225 (действие Swap 225).

Дальнейшее продвижение пакета и изменения MPLS-заголовка происходят аналогичным образом (см. рис 4.74).

4.6. Пример передачи данных в составной сети

В заключение рассмотрим подробный пример, иллюстрирующий процесс формирования протокольных блоков данных на разных уровнях управления передачей данных в составной сети (рис.4.76), использующей стек протоколов TCP/IP.



Составная сеть с помощью трёх маршрутизаторов (Мш1, Мш2, Мш3) объединяет сеть ATM-сеть, X.25 и 5 локальных сетей Ethernet (Eth1, Eth2, Eth3, Eth4, Eth5), к которым подключены пользователи (компьютеры) A, B, C, D, E.

4.6.1. Система обозначений

Введём следующие обозначения

- IP- и MAC-адрес *компьютера*:
  IP.<имя компьютера>
  MAC.<имя компьютера>
- IP- и MAC-адрес *порта маршрутизатора*:
  IP.<номер маршрутизатора>.<номер порта>
  MAC.<номер маршрутизатора>.<номер порта>

- заголовок используемого в сети *кадра* (пакета, ячейки):
  З_<имя сети>

Например:

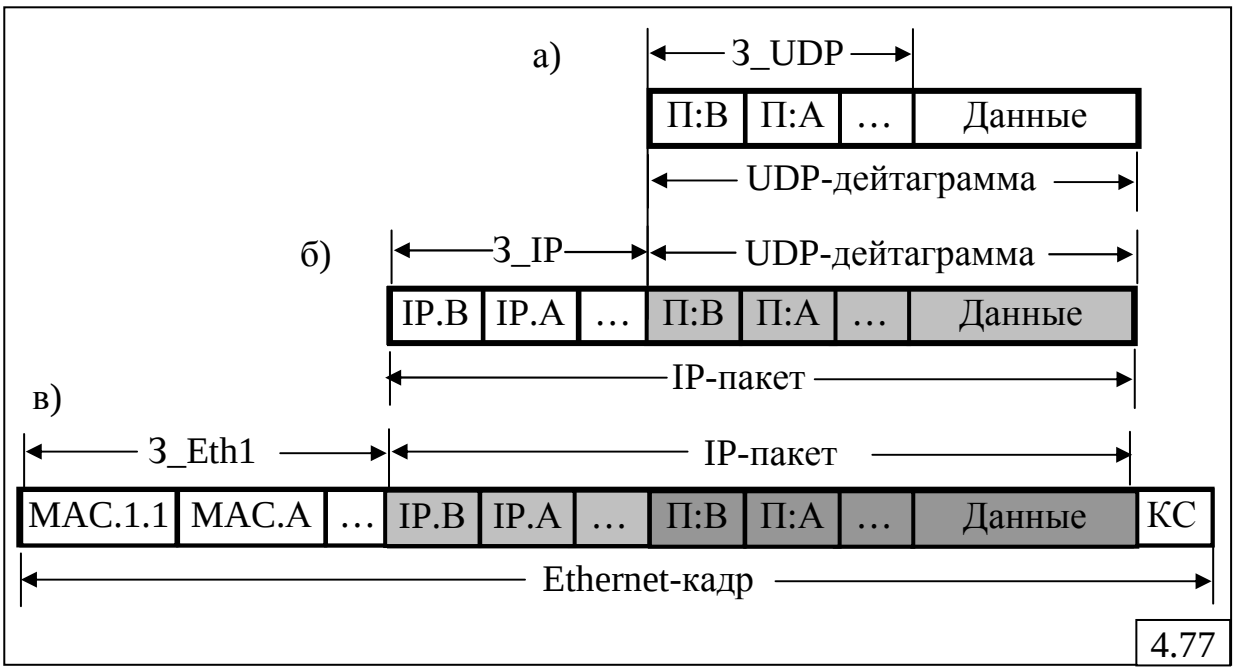
- IP- и MAC-адрес компьютера **А** будут иметь вид:
  IP.A и **MAC.A**,
- IP- и MAC-адрес порта 2 маршрутизатора **Мш1** будут иметь вид:
  IP.1.2 и **MAC.1.2**
- заголовок используемого в сети **Eth1** кадра:
  З_Eth1

Рассмотрим поэтапно, как изменяется протокольный блок данных в зависимости от среды передачи в процессе доставки данных от узла (компьютера) **А** к узлу **В**. Для определённости положим, что для передачи данных из конца в конец используется транспортный протокол UDP.

4.6.2. Формирование данных в узле-источнике

1. Данные, подлежащие передаче, направляются от соответствующего приложения, реализуемого на прикладном уровне в компьютере А, на транспортный уровень, где формируется UDP-дейтаграмма (рис.4.77,а), в заголовке **З_UDP** которой указываются номера двух портов – *получателя* (**П:В**) и *отправителя* (**П:А**).

2. UDP-дейтаграмма передаётся протоколу IP, который вкладывает её в IP-пакет (рис.4.77,б), в заголовке **З_IP** которого указываются IP-адреса *получателя* (**IP.B**) и *отправителя* (**IP.A**).



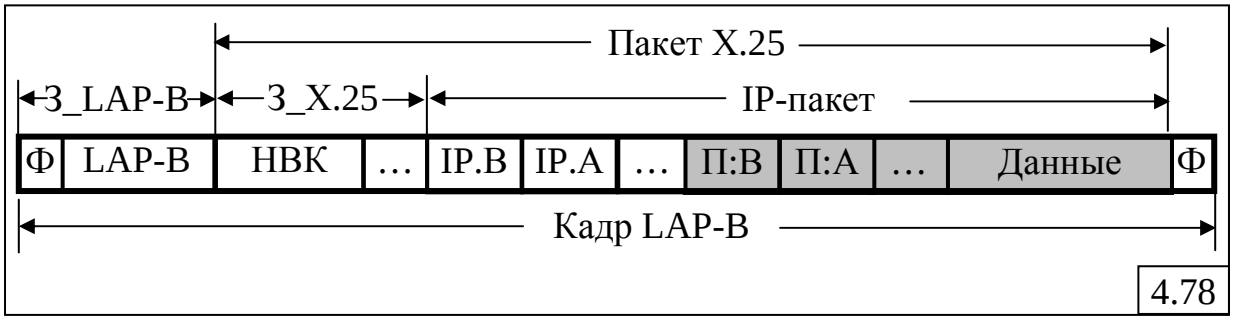
3. IP-пакет поступает на канальный уровень реализуемого в компьютере А стека протоколов, где вкладывается в кадр сети Ethernet, поскольку компьютер А принадлежит сети Eth1. Так как компьютер В не подключён к сети Eth1, компьютер А обращается к таблице маршрутизации и определяет, что для передачи кадра следует

использовать шлюз, которым является маршрутизатор Мш1. Тогда в заголовке кадра **3_Eth1** в качестве MAC-адреса назначения указывается адрес порта 1 маршрутизатора Мш1 – **MAC.1.1**, с которым связана ЛВС Eth1, и MAC-адрес компьютера А – **MAC.A**, являющегося узлом-отправителем кадра (рис.4.77,в). Концевик кадра содержит контрольную сумму (**КС**) для проверки правильности доставки кадра.

4.6.3. Передача данных

4. Сформированный таким образом кадр передаётся на физический уровень, который обеспечивает доставку кадра *от компьютера А через ЛВС Eth1 к маршрутизатору Мш1* в виде физических сигналов (электрических, оптических, ЭПИ), соответствующих среде передачи.

5. Маршрутизатор Мш1, получив кадр, передаёт его для обработки протоколу Ethernet, который подсчитывает контрольную сумму кадра и сравнивает её со значением КС в кадре. Если эти значения не совпадают, то кадр отбрасывается и не записывается в буферную память. В противном случае, если подсчитанное значение контрольной суммы совпадает со значением, указанным в концевике, протокол Ethernet освобождает кадр от заголовка и концевика и передаёт его содержимое, то есть IP-пакет, протоколу IP. Протокол IP анализирует IP-адрес назначения **IP.B** и, используя таблицу маршрутизации, определяет выходной порт и IP-адрес следующего хоста. Положим, что в нашем примере это порт 4 и IP-адрес **IP.3.1**. Поскольку порт 4 маршрутизатора Мш1 связан с сетью X.25 и, следовательно, принадлежит этой сети, протокол IP обращается к протоколу x.25, чтобы с помощью процедуры установления соединения создать виртуальный канал с определённым номером (**НВК**), который заносится в 3-байтовый заголовок пакета X.25. Затем пакет передаётся протоколу канального уровня LAP-B, который вкладывает пакет X.25 в соответствующий кадр LAP-B (рис.4.78), обрамляя его флагами (**Ф**).

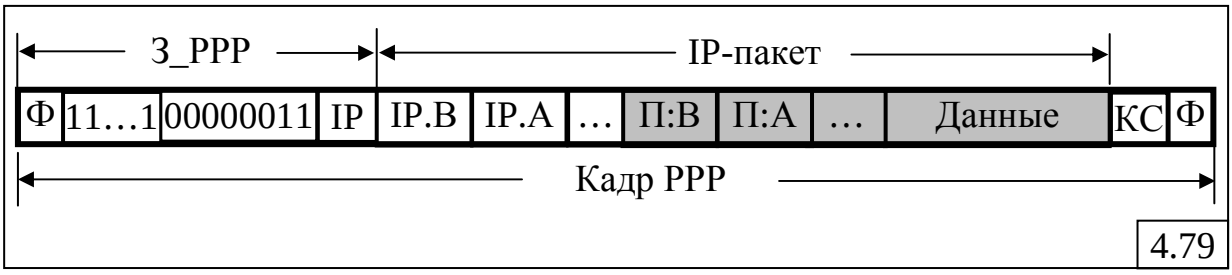


6. Сформированный таким образом кадр передаётся на физический уровень, который обеспечивает доставку кадра *через сеть X.25 от маршрутизатора МШ1 к маршрутизатору Мш3*.

7. Маршрутизатор Мш3, получив кадр, передаёт его для обработки протоколу LAP-B, который освобождает кадр от заголовка **3_LAP-B** и передаёт его содержимое протоколу X.25. Протокол X.25, в свою очередь, извлекает из поля данных пакета X.25 содержимое (IP-пакет) и передаёт его протоколу IP, который, анализируя IP-адрес назначения **IP.B** и

используя свою таблицу маршрутизации, определяет выходной порт и IP-адрес следующего хоста. В нашем примере это порт 2 и IP-адрес **IP.2.5** маршрутизатора Мш2. Поскольку порт 2 маршрутизатора Мш3 напрямую связан с портом 5 маршрутизатора Мш2 выделенным каналом, образуя двухточечное соединение, передача данных осуществляется на основе протокола канального уровня PPP, которому протокол IP передаёт IP-пакет. Протокол PPP вкладывает его в кадр (формат которого показан на рис. 4.69), обрамлённый флагами **Ф** и содержащий три однобайтовых поля:

- **поле адреса (11…1)**, содержащее все единицы;
- **поле «Управление» с кодом 00000011**;
- **поле протокола (IP)**, указывающее, что в поле данных находится IP-пакет (рис.4.79).



8. Сформированный кадр PPP передаётся на физический уровень, который обеспечивает доставку кадра *по выделенному каналу от маршрутизатора МШ3 к маршрутизатору Мш2*.

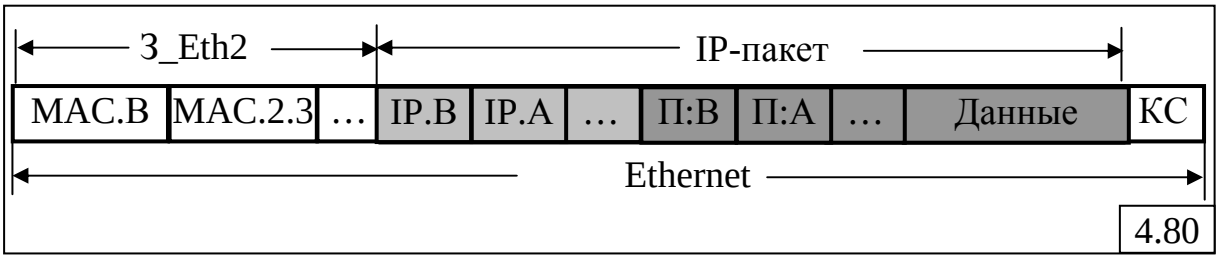
9. Маршрутизатор Мш1, получив кадр PPP, передаёт его для обработки протоколу PPP, который проверяет контрольную сумму и отбрасывает кадр, если рассчитанное значение контрольной суммы не совпадает со значением, указанным в поле КС кадра. Если КС совпадает с указанным в концевике значением, протокол PPP извлекает содержимое, то есть IP-пакет, и передаёт его протоколу IP. Протокол IP анализирует IP-адрес назначения и, используя свою таблицу маршрутизации, определяет, что адресат с IP-адресом **IP.B** находится в локальной сети Eth2, непосредственно подключённой к порту 3 этого маршрутизатора. Затем протокол IP обращается к протоколу ARP, чтобы узнать MAC-адрес, соответствующий IP-адресу IP.B. Протокол ARP находит в своей ARP-таблице MAC-адрес (**MAC.B**) и выдаёт его протоколу IP. Если протокол ARP не находит MAC-адреса, то он реализует процедуру его поиска, посылая в ЛВС Eth2 широковещательный ARP-запрос.

После того как найден MAC-адрес компьютера-получателя B, он вместе с IP-пакетом передаётся протоколу канального уровня Ethernet, который вкладывает его в кадр Ethernet (рис.4.80), указывая в качестве адреса назначения **MAC.B** и адреса отправителя – **MAC.2.3**.

10. Сформированный кадр передаётся на физический уровень, который обеспечивает доставку кадра *через локальную сеть Eth2 от маршрутизатора МШ2 к компьютеру B*.

11. **Компьютер B**, откликаясь на адрес **MAC.B**, записывает поступивший кадр в буфер сетевого адаптера. По завершении приёма кадр

передаётся протоколу Ethernet, который проверяет правильность доставки кадра, извлекает содержимое (IP-пакет) и передаёт его протоколу IP. Последний, в свою очередь, снимает IP-заголовок и передаёт содержимое пакета (UDP-дейтаграмму) протоколу UDP, который в соответствии с указанным в заголовке номером **П:В** порта назначения пересылает содержимое, находящееся в поле данных, конкретному *прикладному процессу*.



4.7. Безопасность компьютерных сетей

Широкое применение компьютерных сетей во всех областях человеческой деятельности, оказывающее существенное влияние на нашу жизнь, делает весьма актуальной проблему информационной безопасности. Защита информации в компьютерных сетях является одной из наиболее важных задач, которые должны решаться в процессе их разработки и эксплуатации.

Средства защиты информации в компьютерных сетях можно разбить на два класса:

- средства *компьютерной безопасности*, обеспечивающие защиту информации, находящейся в локальной сети или на отдельном компьютере пользователя;
- средства *сетевой безопасности*, обеспечивающие защиту информации в процессе её передачи через сеть.

4.7.1. Средства компьютерной безопасности

Средства **компьютерной безопасности** должны обеспечить защиту от несанкционированного доступа всех находящиеся внутри собственной локальной сети ресурсов:

- аппаратных – серверы, дисковые массивы, маршрутизаторы;
- программных – операционные системы, СУБД, почтовые службы и т. п.

Кроме того, необходимо обеспечить защиту данных, хранящихся в файлах и обрабатываемых в компьютерах. Для этого необходимо контролировать трафик, входящий в сеть обычно из Интернета, и стараться перекрыть доступ извне для любой информации, с помощью которой злоумышленник может попытаться использовать внутренние ресурсы сети во вред их владельцу.

Наиболее часто в качестве средства компьютерной безопасности используется брандмауэр, устанавливаемый в местах соединений внутренней локальной сети с Интернетом. **Брандмауэр** (firewall)

представляет собой межсетевой экран, который контролирует трафик между локальной сетью и Интернетом и не пропускает подозрительный трафик в сеть. Кроме того, в качестве средств компьютерной безопасности могут использоваться встроенные средства безопасности операционных систем, баз данных, а также встроенные аппаратные средства компьютера.

4.7.2. Средства сетевой безопасности

Для обеспечения **сетевой безопасности** необходимо защищать информацию, передаваемую в виде пакетов через сети поставщиков услуг Интернета, чтобы она не была искажена, уничтожена или перехвачена посторонними людьми. Для решения этой задачи сегодня широко используется механизм виртуальных частных сетей (VPN).

Автономно работающий компьютер можно более или менее эффективно защитить от внешних покушений. Гораздо сложнее это сделать, если компьютер работает в сети и общается с другими компьютерами. Обеспечение безопасности в этом случае сводится к тому, чтобы сделать проникновение посторонних к ресурсам компьютера контролируемым. Для этого каждому пользователю сети должны быть четко определены его права на доступ к информации, устройствам и на выполнение системных действий в каждом компьютере сети. Дополнительно необходимо обеспечить защиту от перехвата передаваемых по сети данных и создания «ложного» трафика, на что направлена большая часть средств обеспечения сетевой безопасности.

Вопросы сетевой безопасности приобретают особую значимость в связи с тем, что корпоративные сети всё чаще используют Интернет в качестве транспортного средства.

4.7.3. Конфиденциальность, доступность, целостность

Безопасная информационная система должна:

- защищать данные от несанкционированного доступа;
- быть всегда готовой предоставить данные своим пользователям;
- надежно хранить информацию и гарантировать неизменность данных.

Для этого система должна обладать следующими свойствами.

- **Конфиденциальность** (confidentiality) — гарантия того, что секретные данные будут доступны только авторизованным пользователям, которым этот доступ разрешен.
- **Доступность** (availability) — гарантия того, что авторизованные пользователи всегда получат доступ к данным.
- **Целостность** (integrity) — гарантия сохранности данных, которая обеспечивается запретом для неавторизованных пользователей каким-либо образом изменять, модифицировать, разрушать или создавать данные.

Требования безопасности могут меняться в зависимости от назначения системы, характера используемых данных и типа возможных угроз.

Если свойства целостности и доступности актуальны для всех систем, то свойство конфиденциальности может быть необязательным, например, если информация предназначена для широкого круга людей. В то же время, для того чтобы злоумышленник не смог изменить эту информацию, необходимо принять меры по обеспечению целостности данных.

Понятия конфиденциальности, доступности и целостности могут быть применены не только по отношению к информации, но и к другим ресурсам вычислительной сети (внешним устройствам, сетевому оборудованию или приложениям). Конфиденциальность, применительно к какому-либо устройству, обеспечивает доступ к нему только авторизованным пользователям, причем они могут выполнять только те операции, которые им разрешены. Свойство доступности устройства состоит в его готовности к использованию в момент возникновения такой необходимости. Благодаря свойству целостности злоумышленник не сможет изменить параметры настройки устройства, что могло бы привести к выходу его из строя.

4.7.4. Сервисы сетевой безопасности

Для защиты данных используются средства, называемые ***сервисами сетевой безопасности***, которые обеспечивают контроль доступа, включающий процедуры *шифрование информации*, *аутентификации*, *идентификации и авторизации*, *аудит*, *антивирусную защиту*, *контроль сетевого трафика* и т.д. Средства безопасности могут быть либо встроены в программное (операционные системы и приложения) и аппаратное (компьютеры и коммуникационное оборудование) обеспечение сети, либо реализованы в виде отдельных продуктов, созданных специально для решения проблем безопасности.

Рассмотрим основные сервисы сетевой безопасности.

Шифрование — процедура, превращающая информацию из обычного «понятного» вида в «непонятный» зашифрованный вид. Для расшифровки зашифрованной информации используется процедура дешифрирования. Пара процедур – шифрование и дешифрирование – называется **криптосистемой.** Шифрование может применяться в системах аутентификации или авторизации пользователей, а также в системах защиты канала связи и хранения данных.

Аутентификация (от греч. authetikos – подлинный, англ. authentication – опознавание, отождествление) – подтверждение подлинности – предотвращает несанкционированный доступ к сети посторонних лиц и разрешает доступ легальным пользователям.

В качестве объектов, требующих аутентификации, могут выступать не только пользователи, но и различные приложения, устройства, данные.

Примером аутентификации на уровне приложений может служить взаимная аутентификации клиента и сервера, когда клиент, доказавший серверу свою легальность, также должен убедиться, что ведет диалог

действительно со своим сервером. При установлении сеанса связи между двумя устройствами также может быть предусмотрена процедура взаимной аутентификации. Аутентификация данных означает доказательство целостности этих данных, а также факт их поступления именно от того человека, который объявил об этом. Для этого используется механизм **электронной подписи.**

Аутентификацию не следует путать с идентификацией и авторизацией.

Идентификация заключается в сообщении пользователем системе своего идентификатора, в то время как аутентификация — это процедура доказательства пользователем того, что он является тем, за кого себя выдает, в частности доказательство того, что именно ему принадлежит введенный им идентификатор. Идентификаторы пользователей применяются в системе с теми же целями, что и идентификаторы любых других объектов (файлов, процессов, структур данных), и они не всегда связаны непосредственно с обеспечением безопасности.

Авторизация — процедура контроля доступа легальных пользователей к ресурсам системы с предоставлением каждому из них именно тех прав, которые определены ему администратором. Помимо предоставления пользователям прав доступа к каталогам, файлам и принтерам, система авторизации может контролировать возможность выполнения пользователями различных системных функций, таких как локальный доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т. п.

Аудит — фиксация в системном журнале событий, связанных с доступом к защищаемым системным ресурсам. Подсистема аудита современных операционных систем позволяет дифференцированно задавать перечень интересующих администратора событий с помощью удобного графического интерфейса. Средства учета и наблюдения обеспечивают возможность обнаружить и зафиксировать важные события, связанные с безопасностью; любые попытки (в том числе и неудачные) создать, получить доступ или удалить системные ресурсы.

4.7.5. Технология защищённого канала

Технология защищенного канала обеспечивает безопасность передачи данных по открытой транспортной сети, например по Интернету, за счет:

- взаимной аутентификации абонентов при установлении соединения;
- **защиты передаваемых по каналу сообщений от несанкционированного** доступа;
- обеспечения целостности поступающих по каналу сообщений.

Защищенный канал можно построить с помощью протоколов, реализованных на разных уровнях модели OSI (табл.4.6).

Таблица 4.6

Уровни защищаемых протоколов	Протоколы защищенного канала
Прикладной уровень	S/MIME
Уровень представления	SSL, TLS
Сеансовый уровень	
Транспортный уровень	
Сетевой уровень	IPSec
Канальный уровень	PPTP
Физический уровень	

Защита данных средствами верхних уровней (прикладного, представления или сеансового) не зависит от технологий транспортировки данных (IP, Ethernet или ATM), однако приложения зависят от конкретного протокола защищенного канала, так как в них должны быть встроены явные вызовы функций этого протокола. Протоколы безопасности прикладного уровня защищают только вполне определенную сетевую службу, например протокол S/MIME защищает сообщения электронной почты. На уровне представления используется протокол SSL (Secure Socket Layer – слой защищенных сокетов) и его открытая реализация TLS (Transport Layer Security – безопасность транспортного уровня).

Средства защищенного канала становятся прозрачными для приложений в тех случаях, когда они защищают кадры протоколов сетевого и канального уровней. Однако при этом сервис защищенного канала становится зависимым от протокола нижнего уровня.

Компромиссным вариантом защищённого канала является работающий на сетевом уровне протокол IPSec. С одной стороны, он прозрачен для приложений, с другой – может работать практически во всех сетях, так как основан на протоколе IP и использует любую технологию канального уровня (PPP, Ethernet, ATM и т. д.).

4.7.6. Протокол IPSec

IPSec (сокращение от IP Security) – набор протоколов, позволяющих обеспечить защиту данных, передаваемых по межсетевому протоколу IP за счёт подтверждение подлинности и шифрования IP-пакетов. Применение протокола IPSec гарантирует целостность, аутентичность и конфиденциальность данных на протяжении всего пути между двумя узлами сети, который получил название «***защищенный канал»***.

IPSec-протоколы можно разделить на два класса:

- протоколы, отвечающие за защиту потока передаваемых пакетов, к которым относятся два протокола:

ESP (Encapsulating Security Payload — инкапсуляция зашифрованных данных), обеспечивающий шифрацию, целостность и конфиденциальность передаваемых данных;

AH (Authentication Header — заголовок аутентификации), гарантирующий только целостность и аутентичность данных (передаваемые данные не шифруются).

- протокол обмена криптографическими ключами IKE (Internet Key Exchange — обмен ключами Интернета), автоматически предоставляя конечным точкам защищенного канала секретные ключи, необходимые для работы протоколов аутентификации и шифрования данных.

Для шифрования данных в протоколе IPSec может быть применен любой симметричный алгоритм шифрования. В симметричных схемах шифрования конфиденциальность основана на том, что отправитель и получатель обладают общим, известным только им, параметром функции шифрования. Этот параметр называется ***секретным ключом***. Секретный ключ используется как для шифрования текста, так и для его дешифрирования.

Протоколы AH и ESP могут защищать данные в двух режимах:

- транспортном;
- туннельном.

В **транспортном режиме** шифруется только содержимое IP-пакета, не затрагивая заголовок, который не изменяется.

В **туннельном режиме** IP-пакет шифруется целиком, помещается в новый IP-пакет, который передаётся по сети в соответствии с заголовком нового IP -пакета. Таким образом формируется ***защищённый IP-туннель***. Туннельный режим может использоваться для подключения удалённых компьютеров к виртуальной частной сети или для организации безопасной передачи данных через открытые каналы связи (например, Интернет) между шлюзами для объединения разных частей виртуальной частной сети.

Режимы IPSec не являются взаимоисключающими – в одном и том же узле некоторые безопасные соединения могут использовать транспортный режим, а другие — туннельный.

Применение того или иного режима зависит:

- от требований, предъявляемых к защите данных;
- от типа узла, завершающего защищенный канал – хост (конечный узел) или шлюз (промежуточный узел).

Соответственно, имеются три схемы применения протокола IPSec:

- хост—хост;
- шлюз—шлюз;
- хост—шлюз.

В схеме **хост—хост**, использующей, как правило, транспортный режим защиты, защищенный канал устанавливается между двумя конечными узлами сети, и протокол IPSec, работая на конечных узлах, защищает передаваемые данные.

В схеме **шлюз—шлюз,** использующей только туннельный режим защиты, защищенный канал устанавливается между двумя промежуточными узлами, называемыми ***шлюзами безопасности*** (Security Gateway, SG), на каждом из которых работает протокол IPSec. Защищенный обмен данными может происходить между любыми двумя конечными узлами, подключенными к сетям, связанным со шлюзами безопасности. Конечные узлы передают трафик в незащищенном виде, направляя его в общедоступную сеть через шлюз безопасности, который и обеспечивает защиту трафика с помощью протокола IPSec.

Схема **хост—шлюз** применяется при удаленном доступе и позволяет надежно защитить трафик и внутренней сети. Защищенный канал организуется между удаленным хостом, на котором работает протокол IPSec, и шлюзом, защищающим трафик для всех хостов, входящих во внутреннюю сеть.

Протокол AH на приёмной стороне проверяет:

- был ли пакет отправлен тем абонентом, с которым установлено безопасное соединение;
- не искажено ли содержимое пакета;
- не является ли пакет дубликатом уже полученного пакета.

Протокол ESP, кроме перечисленных функций, обеспечивает защиту передаваемых данных от несанкционированного просмотра путем их шифрования.