Глава 2

СЕТИ СВЯЗИ С КОММУТАЦИЕЙ ПАКЕТОВ

2.1. Общие сведения

Существует много технологий построения сетей коммутации пакетов, основными из которых являются Frame Relay (ретрансляция кадров), ATM (режим асинхронной передачи), Internet (Интернет). Достаточно длительная практика эксплуатации этих сетей показала, что сети, построенные на протоколах Интернет, имеют преимущества в простоте, возможностях и стоимости. Этому немало способствовало наличие в Интернет множества хорошо зарекомендовавших себя мультимедийных приложений. Единственный недостаток Интернет – отсутствие достаточных средств транспортировки приложений реального времени, таких как аудио и видео, что преодолевается разработкой специализированных для этих целей протоколов. Поэтому в последнее время, подавляющее большинство производителей оборудования склоняются к реализации сетей следующего поколения (Next Generation Network, NGN) на основе Интернет. Можно сказать, что NGN – это Интернет, дополненная специализированными протоколами для транспортировки различной информации с заданными требованиями, в том числе – в реальном масштабе времени, ибо изначально Интернет проектировалась только как сеть передачи данных.

Взаимодействие сетевых элементов в любой сети происходит при помощи интерфейсов и протоколов. Интерфейс определяет порядок следования информационных сигналов и транспортные (электрические, оптические, радио) свойства для их доставки. Протоколы обмена определяют логические процедуры по их обработке. Описание интерфейсов и протоколов существуют в виде международных стандартов, выпускаемых различными организациями. Для сети нового поколения, основанной на Интернет, описание протоколов публикуются в виде ***запросов для комментариев (Requests for Comments, RFC)***, имеющие различные номера и статус. Решение о публикации RFC принимает Редакция RFC, подчиняющаяся архитектурному совету Интернет (Internet Architecture Board, IAB). Документ, включающий текст, рисунки и иллюстрации, представляется в формате ASCII.

Спецификация протокола или услуги, содержащая в RFC, имеет определенный статус, периодически публикуемый в Internet Official Protocol Standards. Стандартная процедура (Standards Track) присвоения статуса RFC следующая.

Первая ступень – проект стандарта ("Proposed Standard"). Спецификация описывает хорошо известную проблему, требующую разрешения. Внедрение и испытание не обязательно, но желательно. Рабочая группа (Internet Research Steering Group, IESG) может потребовать внедрение и испытание в случае его существенного влияния на другие протоколы. Спецификация не должна иметь существенных технических погрешностей. Операторы должны воспринимать ее как несовершенную спецификацию. Желательно внедрение, тестирование и испытание для его коррекции. Рекомендуется к использованию для заинтересованных операторов. Имеющиеся в главе ссылки на проект стандарта обозначены как (PS – Proposed Standard).

Спецификация, получившая признание нескольких операторов и внедренная как минимум двумя независимыми операторами, при достижении положительных результатов приобретает статус черновика стандарта – "Draft Standard". В случае отсутствия подтверждения успешной демонстрации одной или более опций 2-мя независимыми операторами статус присваивается только при их изъятии из

спецификации. Ответственный за внедрение и проверку – председатель рабочей группы. В протоколе испытаний должна быть отражена работоспособность каждой опции. Рекомендуется к использованию для операторов, использующих соответствующие услуги. Имеющиеся в главе ссылки на черновик стандарта обозначены как (DS – Draft Standard).

Спецификациям, получившим всеобщее внедрение и признание, присваивается статус стандарта – Standard. Рекомендуется к обязательному внедрению. Им присваивается дополнительный ярлык “STDxxx”, однако, они сохраняют свой номер и место в списке RFC. Имеющиеся в главе ссылки на стандарт обозначены как (STD – Standard).

Не все предложения проходят стандартную процедуру. Предложения, которые могут быть использованы в будущем или в текущий момент оказались неопределенными или оказались устаревшими и невостребованными из-за появления более совершенных рекомендаций попадают под нестандартную процедуру (Non-Standards Track). Они приобретают соответствующий статус – Экспериментальный (Experimental), Информационный (Informational) или Исторический (Historic). Имеющиеся в книге ссылки на нестандартные предложения обозначены как (EXP – Experimental, INF – Informational).

Некоторые RFC предназначены для разъяснения пользователям корректного применения стандартных процедур. Им присваивается особый статус Best Current Practice – BCP и дополнительный ярлык “BCPxxx”, они сохраняют свой номер и место в списке RFC.

Сеть нового поколения является широкополосной сетью интегрального обслуживания, предназначенной для коммутации мультимедийной информации, включающей в себя триаду услуг (Triple Service) или их различных комбинаций:

- передача данных (текста, факсимиле, неподвижных изображений);
- передача видео (подвижных изображений);
- передача аудио (речи, музыки).

Основой для реализации триады услуг служат базовые протоколы Интернет.

В Интернет приняты следующие определения.

Узел (Node) – любой сетевой элемент, имеющий хотя бы один сетевой адрес (IP-адрес). Примеры узла: оконечное устройство (хост), маршрутизатор, шлюз.

Хост (Host) – конечный узел, который является либо источником, либо получателем информации, либо тем и другим одновременно. Примеры хоста: компьютер (PC), сервер (Server), принтер (Printer). Как правило, имеет не более одного физического интерфейса.

Маршрутизатор (Router, R) – узел, который ретранслирует пакеты между интерфейсами в соответствии с сетевым адресом. Имеет не менее двух физических интерфейсов. Иногда маршрутизатор называют шлюзом.

Шлюз (Gateway, GW) – узел, связывающий протоколы или каналы связи с различной технологией, например, сеть коммутации каналов и сеть коммутации пакетов.

Модуль (Module) – программно-аппаратное средство, входящее в состав узла и реализующее определенную функцию (протокол). Может одновременно входить в состав другого модуля. Примеры модуля: модуль IP, модуль ICMP.

Интерфейс (Interface) – перечень соглашений между передающей и приемной сторонами об уровне и форме сигналов, размерах сообщений, способах контроля ошибок и т.д., обеспечивающий взаимодействие различных модулей.

Протокол (Protocol) – формальное описание набора правил, которые управляют обменом информацией между модулями равного уровня.

Порт (Port) – 1) интерфейс межсетевого взаимодействия на физическом уровне. Может иметь закрепленный за ним сетевой адрес. Примеры портов: E1, T1, E3, T3. 2)Идентификатор процесса (сетевого приложения) верхнего уровня, который взаимодействует с нижележащими уровнями.

Коммутационное устройство – аппаратное средство, осуществляющее соединение узлов по запросу. Примеры коммутационных устройств: коммутатор (Switch), маршрутизатор (Router).

Локальная сеть (Local Network) – небольшая группа узлов, связанная единым адресным пространством. Примеры локальных сетей: Ethernet, Token Ring.

Пакет (Packet) – структурированная информация, имеющая IP-адрес и полезную нагрузку.

Доступ абонента в Интернет через телефонную сеть общего пользования (ТФОП) показан на рис. 2.1. Абонент автоматической телефонной станции (АТС) должен купить у провайдера сети Интернет (Internet Service Provider – ISP) карту с предоплатой, в которой указан телефонный номер провайдера для доступа в Интернет, имя пользователя (User ID) и пароль (Password). Эти данные абонент должен ввести в персональный компьютер. При установлении модемного соединения на телефонный номер провайдера компьютер соединится с сервером сетевого доступа (Network Access Server, NAS), который запросит у компьютера имя и пароль. Компьютер автоматически вернет ему запрошенную информацию. После этого NAS запросит те же данные (имя и пароль) у сервера аутентификации, авторизации и учета (Authentication, Authorization, Accounting) и сравнит данные имени и пароля, полученные от абонента и от AAA-сервера. В случае их совпадения NAS откроет домашнюю страничку провайдера и начнет обслуживание запросов абонента. Для реализации запроса может потребоваться соединение через магистральную сеть (Backbone Network), которая использует высокоскоростные (от 622 Мбит/с до 1.28 Гбит/с) каналы связи и высокопроизводительные маршрутизаторы (R) для объединения зоновых сетей различных провайдеров.

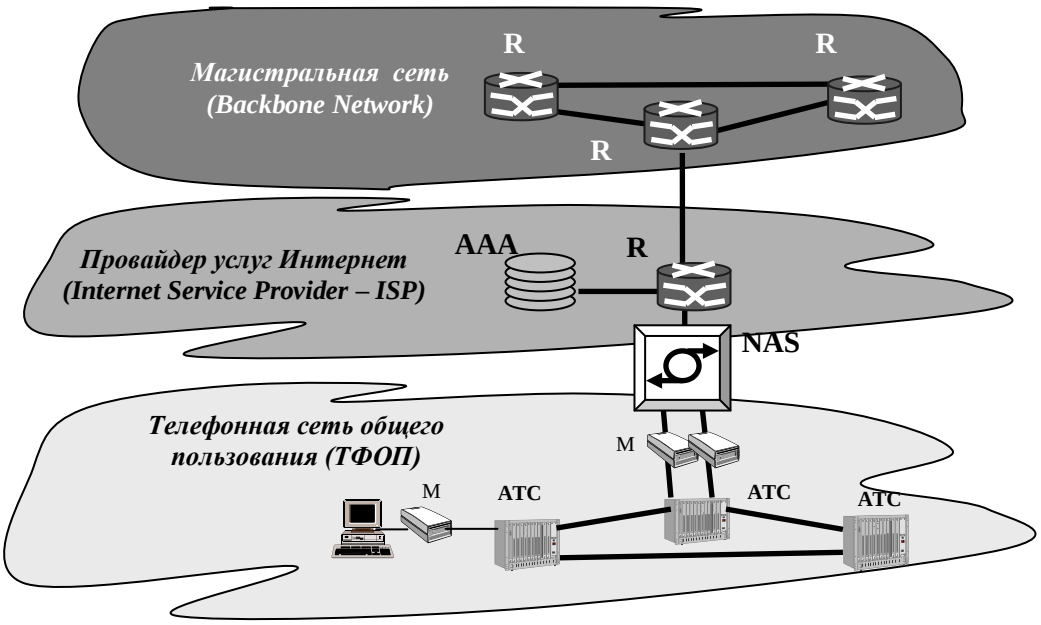

Рис. 2.1. Схема доступа абонента ТФОП в Интернет

ВОПРОСЫ К РАЗДЕЛУ 2.1

1. Дайте определение интерфейса.

Ответ. Перечень соглашений между передающей и приемной сторонами об уровне и форме сигналов, размерах сообщений, способах контроля ошибок и т.д., обеспечивающий взаимодействие различных модулей.

2. Дайте определение протокола.

Ответ. Формальное описание набора правил, которые управляют обменом информацией между модулями равного уровня.

3. Какую информацию хранит AAA сервер?

Ответ. AAA-сервер хранит базу данных пользователей для доступа в Интернет.

2.2. Принципы функционирования Интернет

2.2.1. Модель OSI. Понятие об интерфейсах и протоколах

Организация взаимодействия между элементами сети является сложной задачей, поэтому ее разбивают на несколько более простых задач. Решение каждой простой задачи внутри элемента сети производит модуль, каждый из модулей связан с одним или несколькими другими модулями. Решение сложной задачи подразумевает определенный порядок следования решения простых задач, при котором образуется многоуровневая иерархическая структура (рис. 2.2).

Обмен информацией между модулями происходит на основе определенных соглашений, которые называются интерфейсом. При передаче сообщения модуль верхнего уровня решает свою часть задачи, а его результат, понятный только ему, оформляет в виде дополнительного поля к исходному сообщению и передает измененное сообщение на дообслуживание в нижележащий уровень. Этот процесс называется инкапсуляцией. С другой стороны, при приеме сообщения нижележащий уровень после обработки своей части сообщения удаляет его и оставшееся сообщение передает вышележащему уровню. Таким образом, ***интерфейс*** определяет формат, физические и электрические свойства сигналов обмена между модулями различных уровней, а ***протокол*** – описывает логические процедуры по обработке сообщения удаленному узлу сети равного уровня. ***Рекомендация ITU-T X.200*** описывает семиуровневую модель взаимодействия открытых систем (ВОС, OSI – Open System Interconnection). Наименование уровней и распределение функций между ними следующее.

Физический уровень (Physical Layer) обеспечивает побитную транспортировку кадров (часто называемый пакетом) между узлами по требуемой физической среде передачи (металлический кабель, оптоволоконная линия связи, радиоканал).

На *канальном уровне (Data Link Layer)* реализуются механизмы обнаружения и коррекции ошибок, возникающих в канале связи между узлами. Для этого пакет, поступающий с вышележащего (сетевого) уровня, преобразуется в кадр, т.е. дополняется контрольной суммой и обрамляется специальной последовательностью “Флаг”, позволяющей определить начало и конец кадра. На приеме “Флаги” отбрасываются, и снова вычисляется контрольная сумма. Если вычисленная контрольная сумма совпадает с суммой, принятой из кадра, то кадр считается правильным и в виде пакета передается на сетевой уровень, а на передающую сторону высылается квитирующий кадр. В случае искажения или пропажи кадра квитирующий кадр не высылается и передающая сторона через некоторый промежуток времени возобновляет повторную передачу. Поскольку к узлу (например, маршрутизатору) обычно подключено несколько каналов связи с различными технологиями передачи

кадра, то для каждой технологии передачи канальный уровень добавляет к пакету соответствующее дополнительное поле. Сетевому уровню поставляются пакеты единообразного вида.

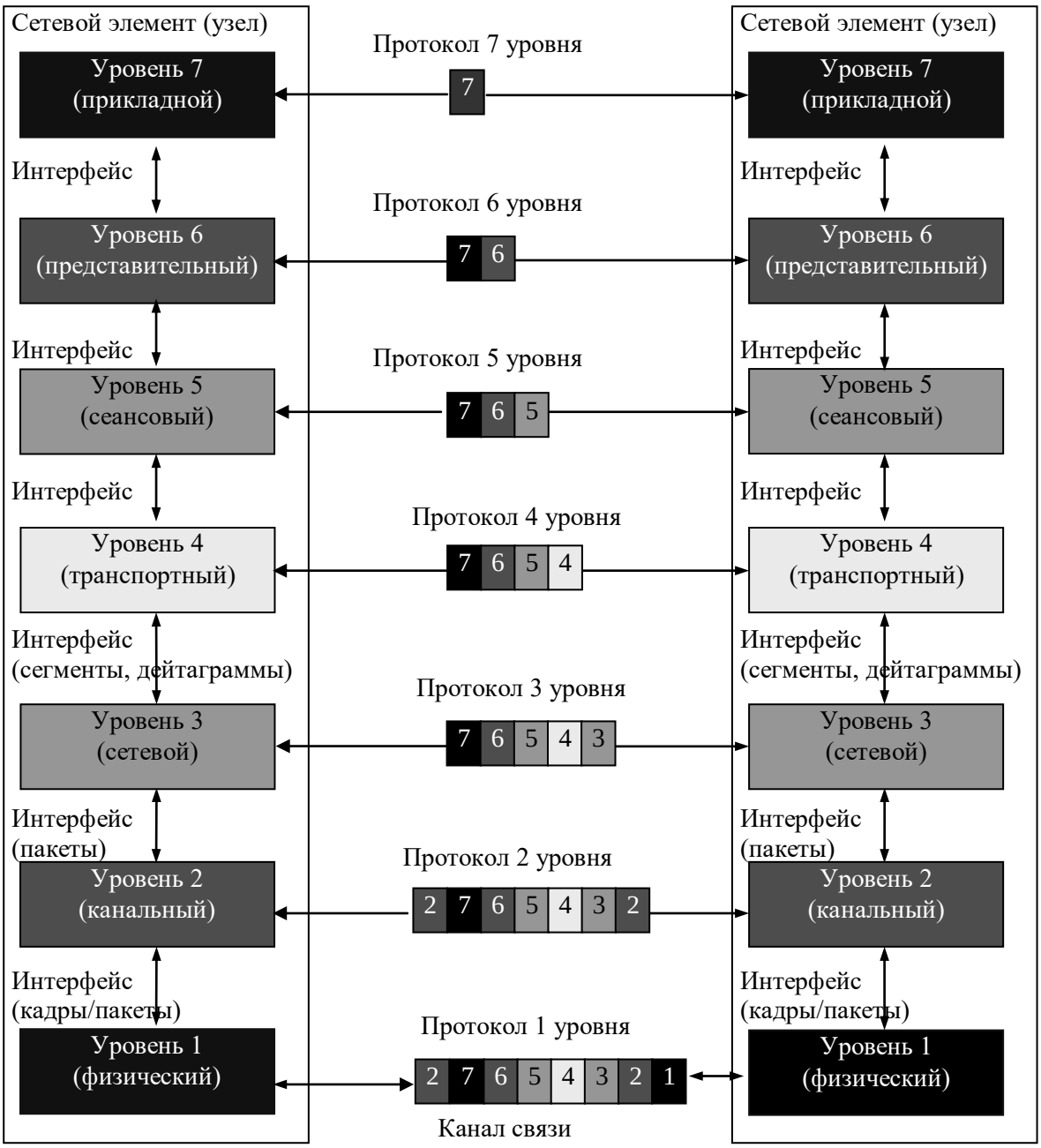

Рис. 2.2. Модель взаимодействия открытых систем (OSI)

Сетевой уровень (Network Layer) служит для образования сквозной транспортной системы между оконечными устройствами пользователя через все промежуточные сети связи – “из конца в конец”. Чтобы передать пакет средства сетевого уровня собирают информацию о топологии сетевых соединений и используют ее для выбора наилучшего пути. Каждый пакет содержит адрес получателя, который состоит из старшей части – номера сети и младшей – номера компьютера (узла) в этой сети. Все компьютеры одной сети имеют один и тот же номер сети, т.е. сеть – это совокупность компьютеров, сетевой адрес которых содержит один и тот же номер сети.

Транспортный уровень (Transport Layer) определяет правила транспортировки пакетов по сети. Эти правила позволяют взаимодействовать с удаленной стороной в двух режимах: не ориентированный на соединение (дейтаграммный режим) и ориентированный на соединение (по предварительно установленному виртуальному соединению). Режим, не ориентированный на соединение (connectionless transfer), транспортный уровень использует для передачи одиночных дейтаграмм, не гарантируя их надежную доставку. Режим, ориентированный на соединение (connection-oriented transfer), используется для надежной доставки данных, при этом каждому сегменту данных присваивается порядковый номер для их правильной сборки на приеме и запускается механизм для обнаружения ошибок и запроса повторной передачи сегмента в случае его утери.

При помощи *сеансового уровня (Session Layer)* организуются диалог между сторонами, фиксируется, какая из сторон является инициатором, какая из сторон активна и каким образом завершается диалог.

Представительный уровень (Presentation Layer) занимается формой предоставления информации нижележащим уровням, например, перекодировкой или шифрованием информации.

Прикладной уровень (Application Layer) это набор протоколов, которыми обмениваются удаленные узлы, реализующие одну и ту же задачу (программу).

Следует отметить, что некоторые сети появились гораздо раньше, чем была разработана модель OSI, поэтому для многих систем соответствие уровней модели OSI весьма условно.

2.2.2. Стек протоколов Интернет

Интернет предназначена для транспортировки любого вида информации от источника к получателю. В транспортировке информации участвуют различные элементы сети – оконечные устройства, коммутационные устройства и серверы. Группы узлов при помощи коммутационных устройств объединяются в локальную сеть, локальные сети соединяются между собой шлюзами (маршрутизаторами). Коммутационные устройства используют различные технологии: Ethernet, Token Ring, FDDI и другие.

Каждое оконечное устройство (хост) может одновременно обслуживать несколько процессов по обработке информации (речь, данные, текст …), которые существуют в виде сетевых приложений (специализированных программ), расположенных на высшем уровне; от приложения информация поступает в средства обработки информации в нижележащие уровни

Транспортировка приложения в каждом узле решается последовательно различными уровнями. Каждый уровень для решения свой части задачи использует свои протоколы и обеспечивает дуплексное прохождение информации. Последовательность прохождения задач образует стек протоколов. В процессе транспортировки информации каждый узел задействует необходимый ему стек протоколов. На рис. 2.3 показан полный стек базовых протоколов сетевого соединения в Интернет.

Узлы, с точки зрения сети, представляют собой источники и получатели информации. Четыре нижних уровня в совокупности независимы от вида передаваемой информации. Каждое сетевое приложение, связывающееся с четвертым уровнем, идентифицируется своим уникальным номером ***порта***. Значения портов занимают диапазон от 0 до 65535. В этом диапазоне номера портов 0-1023 выделены под общесетевые приложения (well-known ports), номера портов 1024-49151 используются разработчиками специализированного программного обеспечения, номера портов

49152-65535 – динамически закрепляются за сетевыми приложениями пользователей на время сеанса связи. Численные значения номеров портов стека приведено в ***RFC 3232*** [77].

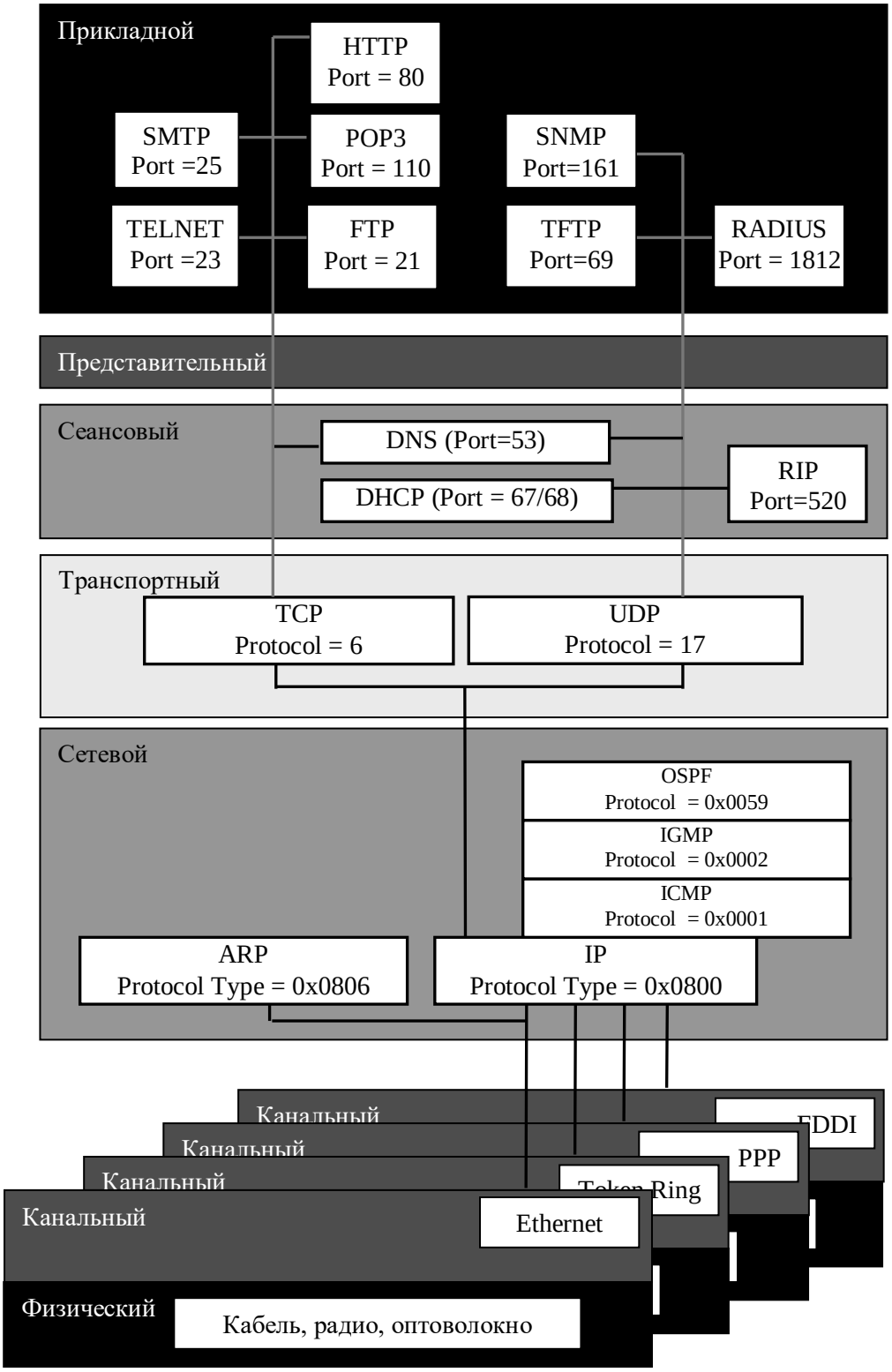

Рис. 2.3. Стек базовых протоколов Интернет

Транспортный (четвертый) уровень поддерживает два режима установления связи – с установлением соединения и без установления соединения. Каждый из режимов идентифицируется своим номером ***протокола (Protocol)***. В стандартах Интернет принято кодирование в шестнадцатеричном коде. Первый режим используется модулем TCP, имеющий код протокола 6 (в шестнадцатеричном коде – 0x06) и используется для гарантированной транспортировки информации. Для этого каждый передаваемый пакет снабжается порядковым номером и должен быть подтвержден приемной стороной о его правильном приеме. Второй режим используется модулем UDP без гарантии доставки информации получателю (гарантия доставки обеспечивается приложением). Протокол UDP имеет код 17 (в шестнадцатеричном коде – 0x11).

Сетевой (третий) уровень обеспечивает перемещение информации в виде пакетов между сетями (интерфейсами канального уровня), используя сетевой адрес. Семейство протоколов третьего уровня нижележащими уровнями идентифицируется ***типом протокола*** (ARP – типом 0x0806 или IP – типом 0x0800)**.** Связка “протокол – сетевой адрес – номер порта ” называется сокетом (socket). Пара сокетов – на передаче и приеме – однозначно определяет установленное соединение. Адрес назначения каждого пакета, поступившего в модуль IP с канального уровня, анализируется, чтобы понять, куда пакет следует дальше направить: в собственное приложение или переместить на другой интерфейс для дальнейшей транспортировке по сети.

Второй (канальный) уровень осуществляет обработку пакетов в локальной сети, используя различные технологии: Ethernet, Token Ring, FDDI и другие.

Первый уровень обеспечивает преобразование бинарных кодов в линейные коды, которые наиболее хорошо подходят к используемой транспортной среде (металлический кабель, оптоволоконная линия связи, радиоканал).

2.2.3. Протоколы доступа в Интернет

Для доступа в Интернет используется семейство протоколов под общим названием ***PPP (Point-to-Point Protocol)***, в число которых входят (***RFC 1661*** [54]):

1. Протокол управления каналом (Link Control Protocol – LCP) для согласования параметров обмена пакетами на канальном уровне на участке хост – сервер сетевого доступа (в частности, для согласования размера пакета и типа протокола аутентификации).

2. Протокол аутентификации (Authentication Protocol) для установления легитимности пользователя (в частности, с использованием протокола Challenge Handshake Authentication Protocol – CHAP).

3. Протокол сетевого управления (IP Control Protocol – IPCP) для конфигурации параметров сетевого обмена (в частности, присвоения IP-адреса).

После этого начинается обмен информацией по протоколу IP.

Каждый из перечисленных протоколов может использовать любую транспортную среду, поэтому существует много способов инкапсуляции PPP на физическом уровне. Для инкапсуляции PPP в каналы связи “точка-точка” используется процедура схожая с HDLC (***RFC 1662*** [55]).

Обмен кадрами с использованием процедуры схожей с HDLC (High-level Data Link Control Procedure) предполагает дуплексный обмен кадрами. Каждый переданный кадр должен быть подтвержден, при отсутствии подтверждения в течение тайм-аута передатчик повторяет передачу.

Структура кадра приведена на рис. 2.4. Порядок передачи полей кадра – слева направо. Назначение полей кадра следующее.

Флаг (Flag) 01111110	Адрес (Address) 11111111	Управление (Control) 00000011	Протокол (Protocol) xxxxxxxx xxxxxxx	Информация (Information) до 1500 байт	Проверочная последовательность (FCS) 2 байта	Флаг (Flag) 01111110

Рис. 2.4. Структура полей HDLC-кадра

Каждый передаваемый кадр должен начинаться и заканчиваться комбинацией "Флаг" (Flag), имеющую битовую структуру вида 01111110 (0x7e). Одна и та же комбинация "Флаг" может быть использована как закрывающая для одного кадра и открывающая для следующего кадра. Комбинации "Флаг" должны выявляться приемной стороной с целью определения границ кадра. Для обеспечения кодонезависимого переноса информации необходимо исключить из последующих полей кадра все комбинации, совпадающие со служебными символами (например, комбинацией "Флаг").

В асинхронном режиме формирование всех полей кадра производится побайтно, каждый байт предваряется битом “старт” и заканчивается битом “стоп”.

В синхронном режиме используется либо байт-вставка, либо бит-вставка. В первом случае в полях кадра производится замена байтовых последовательностей 0x7e (“Флаг”) на 2-байтовые 0x7d и 0x5e, 0x7d на 0x7d и 0x5d, 0x03 на 0x7d и 0x23. Во втором случае после формирования всех полей кадра производится побитовый просмотр содержимого каждого кадра между комбинациями "Флаг" и вставляется бит "ноль" после каждых пяти смежных битов "единица". При декодировании кадра на приеме производится побитовый просмотр содержимого кадра между комбинациями "Флаг" и изъятии бита "ноль" после каждых пяти смежных битов "единица".

Адресное поле (Address) имеет постоянное значение 11111111 (0xff), а поле управления (Control) – значение 00000011 (0x03).

Поле протокола принимает значение 0xc021 для протокола LCP, 0xc223 – для протокола CHAP, 0x8021 – для IPCP и 0x0021 – для протокола IP.

Заполнение информационного поля зависит от типа протокола, но его длина не должна быть менее 4 байт.

Проверочная последовательность (Frame Check Sequence, FCS) на передаче формируется так, чтобы а) при умножении информации между флагами на X^{16} и б) последующем делении по модулю 2 на образующий полином $X^{16} + X^{12} + X^5 + 1$ результат был бы равен постоянному числу 0xf0b8.

Процедура доступа абонента ТФОП в Интернет состоит из нескольких этапов.

На первом этапе используется протокол LCP (Protocol = 0xc021), который использует следующий формат (рис. 2.5).

Flag 0x7e	Address 0xff	Control 0x03	Protocol 0xc021	Code	ID	Length	Type	Length	Data	...	FCS 2 байта	Flag 0x7e

Рис. 2.5. Формат кадра протокола LCP

Поле протокола принимает значение 0xc021. Каждое сообщение характеризуется своим кодом (Code), порядковым номером (ID), длиной (Length). В длину сообщения включаются все поля от Code до FCS. В одном сообщении может содержаться несколько параметров, каждый из которых характеризуется типом параметра (Type), длиной (Length) и данными (Date).

Коды сообщений (Code): 01 – запрос конфигурации (Configure-Request), 02 – подтверждение конфигурации (Configure-Ack), 03 – запрос на смену конфигурации (Configure-Nak), 04 – отказ от конфигурации (Configure-Reject), 05 – запрос на разъединение (Terminate-Request), 06 – подтверждение разъединения (Terminate-Ack).

Полная диаграмма взаимодействия оконечного устройства (Host), сервера сетевого доступа (NAS) и сервера аутентификации, авторизации и учета (AAA) при организации доступа абонента ТФОП в Интернет приведена на рис. 2.6.

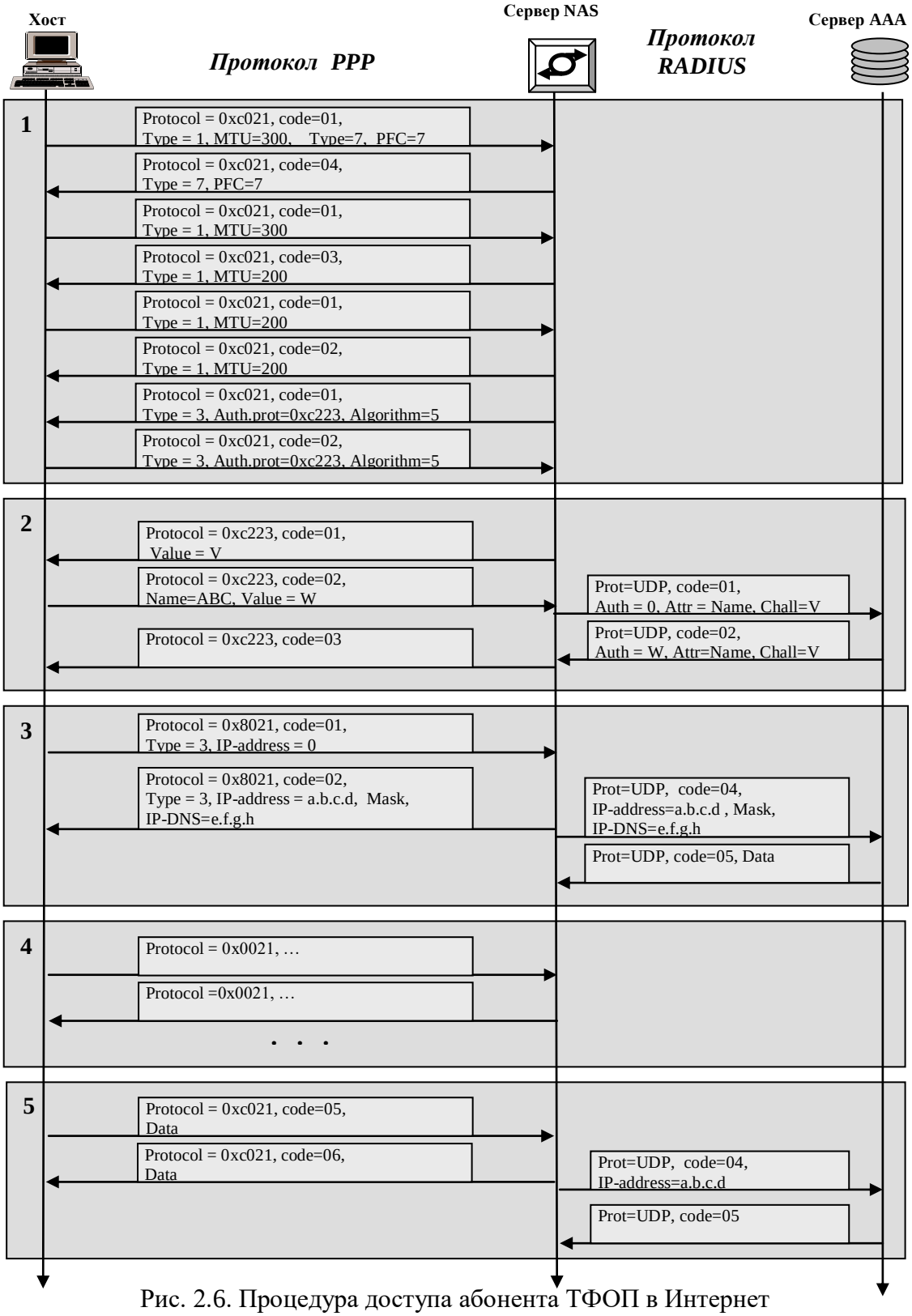

Рис. 2.6. Процедура доступа абонента ТФОП в Интернет

Из рис. 2.6 видно, что вначале хост по протоколу LCP (Protocol = 0xc021) запросил соединение с параметрами MTU=300, PFC=7, но в результате их согласования с сервером доступа NAS (Code=02) установлены параметры MTU=200 (MTU – максимальный размер пакета в байтах), протокол аутентификации – CHAP (Auth.prot=c223). Обмен сжатыми заголовками (PFC=7) сервером доступа NAS был отвергнут (Code=04).

На втором этапе производится аутентификация пользователя с посредством протокола CHAP (Protocol = 0xc223, ***RFC 1994*** [57]). Суть процедуры аутентификации состоит в том, что NAS посылает хосту некоторое случайное число V, а хост возвращает другое число W, вычисленное по заранее известной функции с использованием имени (Name) и пароля (Password), которые вводятся пользователем в компьютер из купленной Интернет карты у провайдера. Иначе говоря, W=f(V, Name, Password). Предполагается, что злоумышленник (хакер) в состоянии перехватить пересылаемые по сети значения V, Name и W, и ему известен алгоритм вычисления функции f. Существо формирования W состоит в том, что исходное элементы (биты) случайного числа V различным образом “перемешиваются” с неизвестным злоумышленнику элементами пароля Password. Затем полученный зашифрованный текст подвергается сжатию, например, суммированию байтов по модулю два. Такое преобразование называется дайджест-функцией (digest function) или хэш-функцией, а результат – дайджестом.

Точная процедура формирования дайджеста определена алгоритмом MD5 и описана в ***RFC 1321*** [49]. NAS по протоколу RADIUS запрашивает у сервера AAA истинное значение W, пересылая ему значения Name и Challenge=V. Сервер AAA на основании полученных от NAS значений V и Name и имеющегося у него в базе данных пароля Password по тому же алгоритму вычисляет W и отсылает его NAS. NAS сравнивает два значения W, полученных от хоста и от сервера AAA: если они совпадают, то хосту посылается сообщение об успешной аутентификации – Success (Code=03).

На третьем этапе происходит конфигурация сетевых параметров согласно ***RFC 1332*** [50] по протоколу IPCP (он же PPP IPC, Protocol=0x8021). Хост запрашивает у NAS сетевые IP-адреса и NAS выделяет из пула (диапазона) IP-адрес для хоста (IP-address=a.b.c.d), а также сообщает IP-адрес DNS-сервера (IP-address=e.f.g.h). NAS по протоколу RADIUS высылает серверу AAA извещение (Code=04) о начале тарификации и получает подтверждение (Code=05).

На 4-м этапе пользователь начинает сеанс связи с Интернет по протоколу IP (Protocol = 0x0021).

После завершения сеанса (этап 5) пользователь по протоколу LCP высылает NAS сообщение о разрушении соединения (Code=05), NAS подтверждает это сообщение (Code=06), отсылает серверу AAA извещение об окончании тарификации и получает от него подтверждение. Все устройства возвращаются в исходное состояние.

2.2.4. Системы счисления и операции, применяемые в IP-сетях

Система счисления состоит из символов и правил их применения. Вес каждого символа зависит от его позиции в изображаемом числе. Счет позиции ведется справа налево, самый правый символ имеет наименьший вес, самый левый – наибольший.

Наиболее распространенной является ***десятичная*** система счисления. Она использует десять символов – 0,1,2,3,4,5,6,7,8,9. Вес первого (самого правого) символа числа умножается на 10^0, следующего – на 10^1 и т.д. Например, десятичное число 3567 представляется как $\mathbf{3}\cdot10^3 + \mathbf{5}\cdot10^2 + \mathbf{6}\cdot10^1 + \mathbf{7}\cdot10^0 = 3567$.

Цифровые вычислительные устройства используют ***двоичную*** систему счисления (двоичный код). Она использует два символа – 0 и 1. Вес первого (самого правого) символа числа умножается на 2^0, следующего – на 2^1 и т.д. Например, десятичное число 3567 в двоичном коде представляется как $\mathbf{1}\cdot 2^{11} + \mathbf{1}\cdot 2^{10} + \mathbf{0}\cdot 2^9 + \mathbf{1}\cdot 2^8 + \mathbf{1}\cdot 2^7 + \mathbf{1}\cdot 2^6 + \mathbf{1}\cdot 2^5 + \mathbf{0}\cdot 2^4 + \mathbf{1}\cdot 2^3 + \mathbf{1}\cdot 2^2 + \mathbf{1}\cdot 2^1 + \mathbf{1}\cdot 2^0 = 110111101111_2 = 2048 + 1024 + 256 + 128 + 64 + 32 + 8 + 4 + 2 + 1 = 3567$.

При описании проколов IP-сетей для сокращения записи вместо двоичного кода используют ***шестнадцатеричную*** систему счисления. Она использует шестнадцать символов – 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Вес первого (самого правого) символа числа умножается на 16^0, следующего – на 16^1 и т.д. Например, десятичное число 3567 в шестнадцатеричном коде представляется как $\mathbf{13}\cdot 16^2 + \mathbf{14}\cdot 16^1 + \mathbf{15}\cdot 16^0 = \mathbf{D}\cdot 16^2 + \mathbf{E}\cdot 16^1 + \mathbf{F}\cdot 16^0$ = 0xDEF = 0xdef, где символ “0x” обозначает запись числа в шестнадцатеричном коде. Разницы между заглавными и прописными символами нет.

Соответствие между значениями символов в десятичной, двоичной и шестнадцатеричной системе счисления приведено в табл. 2.1.

Табл. 2.1 – Соответствие кодов различных систем счисления

Десятичный код	Двоичный код	Шестнадцатеричный код
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Для коммутации в IP-сетях используют IP-адрес, состоящий из 32 двоичных символов (битов). Для удобства работы это двоичное число разбивают на 4 группы по 8 битов, каждую группу представляют в десятичном коде и отделяют одну группу от другой точкой. Такой способ представления IP-адреса называется ***десятично-точечным*** представлением. Например, пусть Y1=73, Y2=12, Y3=85 и Y4=167. Тогда десятично-точечное представление IP-адреса Y1.Y2.Y3.Y4 в двоичном и шестнадцатеричном коде имеет вид табл. 2.2.

Табл. 2.2 – Различное представление IP-адреса

Десятично-точечное представление IP-адреса	73.	12.	85.	167
IP-адрес в двоичном коде	01001001	00001100	01010101	10100111
IP-адрес в шестнадцатеричном коде	49	0C	55	A7

Для шифрования информации в IP-сетях часто применяют операцию вычисления числа M по модулю n, которое записывается в виде W = M mod n и находится как остаток от деления M/n. Например, пусть Y1=73, Y2=12, Y3=85, Y4=167 и требуется

найти значение W = Y1Y2Y3Y4 mod 5 = 731285167 mod 5. Рекомендуется следующий алгоритм вычисления W на калькуляторе.

1. Находим частное от деления w1=731285167/5=146257033.4;
2. Находим целую часть частного w2=INT(w1)=146257033;
3. Находим произведение w3=5·w2=731285165;
4. Находим искомое W=735885167mod 5 = 731285167 – w3 = 2.

При вычислении сетевого IP-адреса маршрутизаторы используют логическую операцию “И” поразрядного сложения чисел. Правило сложения разрядов с помощью логической операции “И”: 0&0=0, 0&1=0, 1&0=0, 1&1=1. Например, результатом логического сложения по “И” чисел Y1=73, Y2=240 будет

Y1= 89 ➔ 01011001_2
&Y2= 240 ➔ 11110000_2

01010000_2 ➔ 80.

При сравнении сетевых IP-адресов маршрутизаторы используют логическую операцию поразрядного сложения чисел по модулю 2 (XOR). Правило сложения разрядов по модулю 2: 0^0=0, 0^1=1, 1^0=1, 1^1=0. Например, результатом логического сложения по модулю 2 чисел Y1=73, Y2=73 будет

Y1= 73 ➔ 01001001_2
^Y2= 73 ➔ 01001001_2

00000000_2 ➔ 0.

2.2.5. IP-адресация версии 4

Установление соединения между двумя и более узлами происходит на основе обработки адресной информации, которую по мере необходимости обрабатываются устройствами 3-го уровня в маршрутизаторах. К адресу предъявляются следующие требования [3,4].

- Адрес должен быть универсальным.
- Адрес должен иметь иерархическую структуру, удобную для обработки соответствующими узлами.
- Адрес должен быть удобен для пользователя.

В связи с этим в Интернет используются три типа адреса.

Физический адрес (Media Access Control, MAC-адрес) используется для установления соединения в локальной сети (подсети). Этот адрес совпадает с номером сетевого адаптера (сетевой карты) компьютера и жестко устанавливается заводом-изготовителем из пула (диапазона), отведенных ему адресов. Записывается в виде шестнадцатеричных чисел, разделенных двоеточием, например, 08:00:06:3F:D4:E1, где первые три значения определяют фирму-производителя (00:10:5a:xx:xx:xx – 3Com, 00:03:ba:xx:xx:xx – Sun, 00:01:e3:xx:xx:xx – Siemens), а последующие – порядковый номер узла. Компьютер может иметь несколько сетевых карт и, соответственно, несколько MAC-адресов. При замене аппаратуры изменяется и MAC-адрес, поэтому их использование в качестве сетевых адресов неудобно.

Символьные (доменные) адреса предназначены для людей. Для работы в больших сетях символьные адреса имеют сложную иерархическую структуру, содержащую имя пользователя, имя подсети (поддомена), символьное имя страны или организации (домена). Например, адрес Ivan.Sidorov@sk.sut.ru обозначает, что адресат (Иван Сидоров) находится в подсети sk сети sut в России (ru), а адрес www.protocols.com – адрес домена коммерческой организации.

Сетевые адреса (***IP-адреса***) фиксированной длины применяются для установления сетевых соединений в Интернет, т.к. использование символьных имен из-за их переменной длины неудобно.

Между физическим адресом (MAC-адресом), символьным (доменным) именем и IP-адресом нет никакого аналитического соответствия, поэтому приходится использовать специальные таблицы соответствия, которые хранятся в специальных сетевых устройствах (серверах). Присвоение IP-адресов, установление соответствия между различными типами адресов осуществляется посредством определенных протоколов.

Согласно ***RFC 3232*** [77] в версии 4 сетевые IP-адреса имеют двухуровневую иерархию, старшая часть которых отображает номер сети, а младшая – номер узла (компьютера) в сети. Общая длина адреса имеет длину 4 байта и записывается в виде десятичных чисел, разделенных точками. Первые биты сетевого адреса задают класс адреса, по которому определяется, какая его часть относится к номеру сети, а какая – к номеру узла. Если сеть является частью Интернет, то номер сети назначается централизованно по рекомендации специального органа Интернет – Internet Information Center, IIC. Номер узла в IP-адресе назначается из разрешенного для этого класса диапазона независимо от физического адреса. Маршрутизатор объединяет несколько сетей, поэтому каждый порт (интерфейс) маршрутизатора имеет свой IP-адрес.

В версии IPv4 имеется пять классов адресов, приведенных в табл. 2.3, где жирным шрифтом выделена старшая часть IP-адреса, указывающая номер сети.

Табл. 2.3 – Классы IP-адресов

Класс	Первые биты IP-адреса	Наименьший **номер сети**	Наибольший **номер сети**	Максимальное число сетей	Макс. число узлов в каждой сети
A	0	**0.**0.0.0	**127.**0.0.0	$2^{7}-2$	$2^{24}-2$
B	10	**128.0.**0.0	**191.255.**0.0	$2^{14}-2$	$2^{16}-2$
C	110	**192.0.0.**0	**223.255.255.**0	$2^{21}-2$	$2^{8}-2$
D	1110	**224**.0.0.0	**239.**255.255.255		Групповые адреса
E	1111	**240**.0.0.0	**255**.255.255.255		Эксперим.+Резерв

Большие сети используют адреса класса A, средние – класса B, маленькие – класса C. В IPv4 существуют определенные соглашения об использовании адресов.

1. Сеть с номером 0.0.0.0 зарезервирована для использования в служебных сообщениях, а сеть с номером 127.0.0.0 используется для петлевого соединения (пересылки пакетов самим себе), поэтому общее количество сетей класса A равно 126.

2. Маршрутизация пакета в публичной сети всегда производится на основании классического IP-адреса номера сети, согласно табл. 2.3, поэтому адрес сети не может быть назначен ни одному узлу.

3. Адрес узла со всеми двоичными “1” предназначен только для адресации всем узлам соответствующей сети (широковещательная рассылка), поэтому этот адрес не может быть назначен ни одному узлу. Совместно с пунктом 2 это означает, что число узлов в любой сети уменьшается на 2.

4. В каждом классе имеется ***диапазон сетевых адресов для частного использования***, которые в публичных сетях отсутствуют. Они используются для построения локальных корпоративных сетей. В классе A – это сеть 10.0.0.0, в классе B – диапазоны сетей от 172.16.0.0 до 172.31.0.0, в классе C – диапазон сетей от 192.168.0.0. до 192.168.255.255.

5. Основное назначение адресов класса D – распространение информации по схеме “один-ко-многим” для групповой рассылки в Интернет аудио- и видеоинформации. Хост, который хочет осуществить рассылку, с помощью протокола группового обслуживания Интернет (Internet Group Management Protocol – IGMP) сообщает о создании в сети мультивещательной группы с определенным адресом.

Устройства, которые хотят присоединиться к создаваемой группе, высылают свое подтверждение. Одно и то же устройство может входить в несколько групп, в одну и ту же группу могут входить устройства различных сетей.

6. Адреса класса E зарезервированы для будущих применений.

Маршрутизация пакета в публичной сети всегда производится на основании IP-адреса номера сети, согласно табл. 2.3. Номер сети принято обозначать с помощью маски, количество лидирующих “единиц” в маске показывает число старших разрядов, которые определяют номер сети. Запись маски производится в формате IP-адреса. Таким образом, для сети класса A стандартная маска имеет вид 255.0.0.0 (в двоичном коде 11111111.00000000.00000000.00000000), для сети класса B – 255.255.0.0 (11111111.11111111.00000000.00000000), для сети класса C – 255.255.255.0 (11111111.11111111.11111111.00000000).

Наличие только четырех классов адресов часто бывает неудобно. Например, администратору необходимо создать сеть из 8000 узлов. Сеть класса C (254 узла) слишком мала, а сеть класса B (65534) слишком велика. Эта проблема решается с помощью создания подсетей, путем переназначения части битов узла в качестве битов сети. Процесс заимствования части битов всегда начинается с крайнего левого бита.

Пусть, например, для нашего случая администратор получил адрес сети 135.38.0.0 (адрес класса B, маска сети 255.255.0.0). В этой сети 16 битов выделено под адрес сети и 16 битов – на адрес узла. Администратору необходимо иметь 8000 узлов, на это необходимо выделить только 13 битов на адреса узлов (2^{13} = 8192), следовательно, оставшиеся 16 – 13 = 3 бита адреса узла можно переназначить как адрес сети. Тогда маска образованной подсети в двоичном коде будет иметь вид 11111111.11111111.**111**00000.00000000 или 255.255.224.0 (жирным шрифтом выделены заимствованные биты адреса узлов класса B). В результате такого деления получим следующие адреса подсетей (табл. 2.4).

Табл. 2.4 – Деление сети на подсети

Номер сети				Число узлов в подсети
~~**10000111 135.**~~	~~**00100110 38.**~~	~~**000**00000 **0.**~~	~~00000000 0~~	~~8190~~
10000111 135.	**00100110 38.**	**001**00000 **32.**	00000000 0	8190
10000111 135.	**00100110 38.**	**010**00000 **64.**	00000000 0	8190
10000111 135.	**00100110 38.**	**011**00000 **96.**	00000000 0	8190
10000111 135.	**00100110 38.**	**100**00000 **128.**	00000000 0	8190
10000111 135.	**00100110 38.**	**101**00000 **160.**	00000000 0	8190
10000111 135.	**00100110 38.**	**110**00000 **192.**	00000000 0	8190
~~**10000111 135.**~~	~~**00100110 38.**~~	~~**111**00000 **224.**~~	~~00000000 0~~	~~8190~~

Две полученные подсети 135.38.0.0 и 135.38.224.0 использовать не рекомендуется, т.к. некоторые типы маршрутизаторов не поддерживают подсети, сетевой адрес которых совпадает с адресом исходной классической сети (135.38.0.0) или с адресом широковещательной рассылки исходной классической сети (135.38.255.255). Теперь одну из оставшихся 6 подсетей (например, подсеть 135.38.32.0) администратор

использует для своих нужд, а оставшиеся пять сетей может отдать другому администратору.

Архитектура местоположения подсети 135.38.32.0 приведена на рис. 2.7.

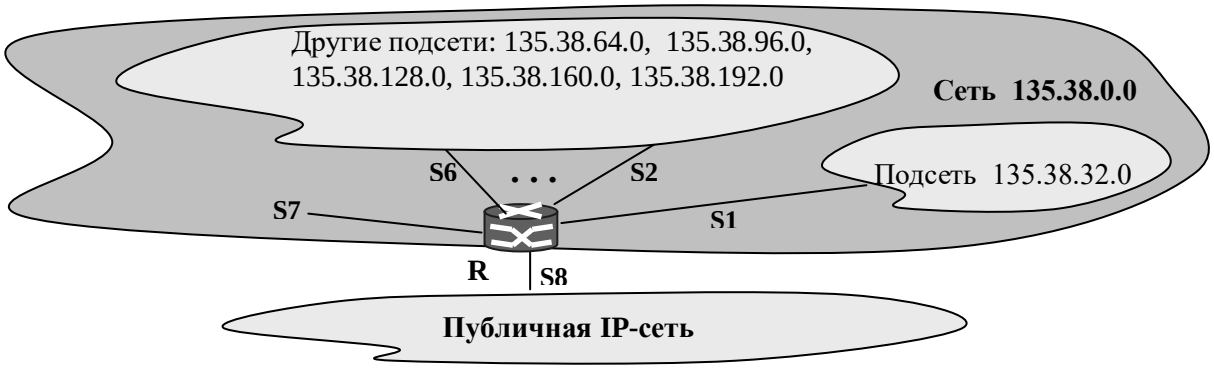

Рис. 2.7. Архитектура местоположения подсети 135.38.32.0

Для обслуживания пакетов маршрутизатор R (рис. 2.7) использует таблицу маршрутов (табл. 2.5).

Табл. 2.5 – Таблица маршрутов

Пункт назначения (Destination)	Маска сети (Subnet Mask)	Пункт пересылки (Next Hop)	Интерфейс (Interface)	Метрика (Metric)
135.38.0.0	255.255.224.0	135.38.0.2	s7	60
135.38.32.0	255.255.224.0	135.38.32.2	s1	1
135.38.64.0	255.255.224.0	135.38.64.2	s2	1
135.38.96.0	255.255.224.0	135.38.96.2	s3	1
135.38.128.0	255.255.224.0	135.38.128.2	s4	1
135.38.160.0	255.255.224.0	135.38.160.2	s5	1
135.38.192.0	255.255.224.0	135.38.192.2	s6	1
135.38.224.0	255.255.224.0	135.38.0.2	s7	60
Default		0.0.0.0	s8	10

Таблица маршрутов в общем случае содержит следующие колонки.

1. Пункт назначения (Destination) – определяет IP-адрес сети назначения.
2. Маска сети (Subnet Mask) – задает количество лидирующих бит в IP-адресе, которые определяют число бит адреса сети.
3. Пункт пересылки (Next Hop) – задает IP-адрес интерфейса следующего маршрутизатора, на который следует направить поступивший пакет (в данном случае совпадает с собственным IP-адресом выходного интерфейса).
4. Интерфейс (Interface) – задает выходной интерфейс, на который следует направить поступивший пакет для достижения пункта пересылки (Next Hop).
5. Метрика (Metric) – задает предпочтение в выборе альтернативных маршрутов. Маршруты с меньшей метрикой более предпочтительны.

Для определения дальнейшего маршрута следования поступившего пакета маршрутизатор производит следующие операции.

1. Поступивший IP-адрес в двоичном коде с помощью логической операции “И” складывается поразрядно с маской сети первой строки таблицы маршрутизации. Правило сложения двоичных разрядов с помощью логической операции “И”: 0 & 0=0, 0 & 1=0, 1 & 0=0, 1 & 1=1.

2. Полученный в результате сложения адрес сети сравнивается с IP-адресом пункта назначения первой строки. При их совпадении поступивший пакет направляется на интерфейс s1.

3. В случае несовпадения те же операции, начиная с пункта 1, проделываются с последующими строками маршрутной таблицы, если они имеются.

4. Все поступившие пакты, адреса которых не найдены в таблице маршрутов, по умолчанию направляются на интерфейс s8.

Например, пусть из подсети 135.38.32.0 по интерфейсу s1 поступили следующие пакеты с IP-адресами назначения: 135.38.16.15, 135.38.56.211, 135.38.92.10, 136.38.92.10. На какие интерфейсы они будут направлены?

Для определения номера адресуемой сети складываем по “И” IP-адрес назначения первого пакета с маской сети, получим

IP=135.38.16.15 → 10000111 00100110 00010000 00001111&
Mask= 255.255.224.0 → 11111111 11111111 11100000 00000000

Destination = 10000111 00100110 00000000 00000000 → 135. 38.0.0.

Пакет будет направлен на не активный интерфейс s7. Поэтому маршрутизатор высылает сообщение ICMP типа 3 (недоступность сети).

Для второго пакета

IP=135.38.56.211 → 10000111 00100110 00111000 11010011&
Mask= 255.255.224.0 → 11111111 11111111 11100000 00000000

Destination = 10000111 00100110 00100000 00000000 → 135. 38.32.0.

Получен адрес собственной подсети. Высылается сообщение ICMP типа 5.

Для третьего пакета

IP=135.38.92.10 → 10000111 00100110 01011100 00001010&
Mask= 255.255.224.0 → 11111111 11111111 11100000 00000000

Destination = 10000111 00100110 01000000 00000000 → 135. 38.64.0.

Пакет будет направлен на интерфейс s2.

Для четвертого пакета

IP=171.38.92.10 → 10101011 00100110 01011100 00001010&
Mask= 255.255.224.0 → 11111111 11111111 11100000 00000000

Destination = 10101011 00100110 01000000 00000000 → 171. 38.64.0.

Пакет будет направлен на интерфейс s8.

Для сокращения числа записей в маршрутизаторе широко используют бесклассовую систему междоменной маршрутизации (Classless Inter Domain Routing, CIDR), в которой маршрутизация осуществляется на основе ***префиксов***. Префиксом обозначают общую часть старших разрядов адреса, которая учитывается при маршрутизации. Они позволяют сократить число записей в маршрутизаторе. Так, если две первые указанные подсети подсоединены к одному маршрутизатору, то смежным маршрутизаторам не обязательно иметь 2 записи о каждой подсети, а достаточно иметь одну запись префикса 135.38.0.0/17 с маской 255.255.128.0 (11111111 11111111 10000000 00000000), содержащей 17 лидирующих “единиц”.

2.2.6. IP-адресация версии 6

Принципиальным решением устранения дефицита сетевых адресов является расширение адресного поля с 4-байтного (текущей версии IPv4) на 16-байтное (в новой версии IPv6). Адреса присваиваются интерфейсам; один интерфейс может иметь

несколько IP-адресов. Адресация использует бесклассовую систему междоменной маршрутизации (Classless Inter Domain Routing, CIDR).

В ***RFC 4291*** [83] предложено три формы представления адреса.

- Предпочтительная форма – восемь 16-битовых шестнадцатиричных чисел, разделенных двоеточием. Например, FE80:0000:0000:0000:3656:78FF:FE9A:BCDE.
- Сжатая форма – запись длительной последовательности “0” путем введения двойного двоеточия. Двоеточие допускается использовать только в одном месте адреса. Например, адрес FE80:0000:0000:0000:3656:78FF:FE9A:BCDE может быть представлен в виде FE80:: 3656:78FF:FE9A:BCDE.
- Смешанная форма (IPv4 Mapping fddress) – шесть старших чисел (96 бит) записываются в предпочтительной форме, а младшие числа (32 бита) представляются в виде, принятом в IPv4. Например, 0:0:0:0:0:FFFF:13.1.68.3 или в сжатой форме ::ffff:13.1.68.3/96..

Длина префикса записывается в конце адреса и отделяется от него косой линией. Например, адрес **12AB:0000:0000:CD3**0:0000:0000:0000:0000/60 содержит 60 битовый префикс 12AB:0000:0000:CD3 или в сжатой форме **12AB:0:0:CD3**0::/60.

В общем случае адрес представляется в виде рис. 2.8.

Subnet prefix (Network ID)	Interface Identifier (Interface ID)
Префикс подсети (n бит)	Идентификатор интерфейса (128-n бит)

Рис. 2.8. Формат адреса в IPv6

Для облегчения функции маршрутизации индивидуальный адрес формируются в виде составного адреса (Aggregatable Global Unicast Address), старшая часть адреса сети используется магистральными маршрутизаторами, а младшая – маршрутизаторами низшего уровня. В совокупности старшие разряды адреса образуют адрес сети/подсети, а младшие – номер интерфейса в этой подсети.

В версии IPv6 имеется три типа адресов.

Индивидуальные адреса (Unicast). Пакет, посланный по такому адресу, доставляется на указанный интерфейс. Хост имеет несколько таких алресов:

0:0:0:0:0:0:0:1 (loopback,) – петлевой адрес, используется для посылки пакета самому себе;

fe80:0:0:0:x:x:x:x (link-local address, префикс fe80::/10) – локальный адрес узла. Хост присваивает его самостоятельно следуя процедурам ***RFC 2464*** [69]. В середину имеющегося MAC-адреса (например, 34:56:78:9a:bc:de) вставляется комбинация **fffe**, значение первого байта увеличивается на 2. В результате получается автоконфигурируемый идентификатор интерфейса 3***6***56:78**ff**:**fe**9a:bcde и локальный адрес узла fe80:0:0:0:3656:78ff:fe9a:bcde. Этот адрес действует только в пределах локальной сети и не обрабатывается маршрутизаторами глобаьной сети, но позволяют организовывать локальные сети без провайдера Internet.

2xxx:x:x:x:x:x:x:x (выделенные в настоящее время глобальные адреса, префикс 2000::/3). Эти адреса присваиваются централизованно, чтобы исключить дублирование IP-адресов.

Выборочный групповой адрес (Anycast). Адрес такого типа используется только маршрутизаторами. Пакет, посланный по такому адресу, доставляется одному ближайшему интерфейсу, маршрут к которому имеет наименьшую метрику. Предполагается для транспортировки пакета по фиксированному пути путем создания стека выборочных адресов. Формат адреса содержит префикс адреса длиной N бит, завершаемый последовательностью “0” длиной (128 – N) бит.

Групповой адрес (Multicast). Адрес такого типа относится к различным узлам. Пакет, посланный по такому адресу, доставляется всем интерфейсам, ассоциированным с этим типом адреса. Формат адреса содержит начальный байт FF (префикс FF00::/8), полубайт типа группы (0 – постоянная, 1 – временная), полубайт области действия (1 – всем интерфейсам, 2 – всем маршрутизаторам локальной сети, …), завершаемый групповым идентификатором длиной 112 бит.

Имеются специальные групповые адреса: FF02:0:0:0:0:0:0:1 (FF02::1) – всем интерфейсам локальной сети, FF02:0:0:0:0:0:0:2 (FF02::2) – всем маршрутизаторам локальной сети.

2.2.7. Присвоение IP-адресов

В локальной сети присвоение пользователям IP-адресов производится с помощью протокола динамической конфигурации компьютера (***Dynamic Host Configuration Protocol, DHCP***) в соответствии с моделью клиент-сервер. При включении, компьютер-клиент посылает в сеть широковещательный UDP-запрос на закрепление IP-адреса за MAC-адресом компьютера. Находящийся в той же сети DHCP-сервер в ответ высылает статический или динамический IP-адрес и IP-адрес DNS-сервера .

Статический IP-адрес выдается клиенту в постоянное пользование из пула (диапазона) адресов, задаваемых администратором при конфигурировании DHCP-сервера. Между физическим и сетевым адресом имеется жесткое соответствие, при всех запросах DHCP-сервер возвращает клиенту один и тот же IP-адрес.

При ***динамическом*** распределении сетевой IP-адрес выдается клиенту на ограниченное время пользования (на время аренды) из пула адресов, задаваемых администратором при конфигурировании DHCP-сервера. До истечения срока аренды между физическим и сетевым адресом имеется жесткое соответствие, при всех запросах DHCP-сервер возвращает клиенту один и тот же IP-адрес. По истечении срока аренды (например, сеанса связи) тот же IP-адрес может быть присвоен другому компьютеру, что позволяет провайдерам строить IP-сеть с количеством компьютеров, превышающих число IP-адресов.

Для экономии внешних сетевых IP-адресов часто используют ***транслятор сетевых адресов (Network Address Port Translation, NAPT)***. В этом случае, при установлении связи из частной сети во внешнюю сеть, маршрутизатор подменяет адрес частного использования (например, 10.0.0.5) на один из нескольких или даже на один и тот же внешний IP-адрес (например, 208.106.44.5), но с уникальным для каждого сеанса связи номером внешнего порта маршрутизатора.

2.2.8. Принцип коммутации в IP-сети

Рассмотрим процессы установления соединений в IP-сети рис. 2.9 между двумя локальными сетями X и Y. В сети X имеется два хоста: хост A, имеющий IP-адрес IPA, и хост B, имеющий IP-адрес IPB; в сети Y также имеется два хоста: C с IP-адресом IPC и хост D с IP-адресом IPD. Сети напрямую соединены через маршрутизаторы R1 и R2. Дополнительно может быть использован путь через маршрутизатор R3.

Предположим, что пользователь A запускает некий прикладной процесс, требующий установления соединения с хостом B своей же сети. От приложения узла A в модуль IP поступает дейтаграмма с IP-адресом назначения IPB. Модуль IP хоста A имеет следующую таблицу маршрутизации (табл. 2.6).

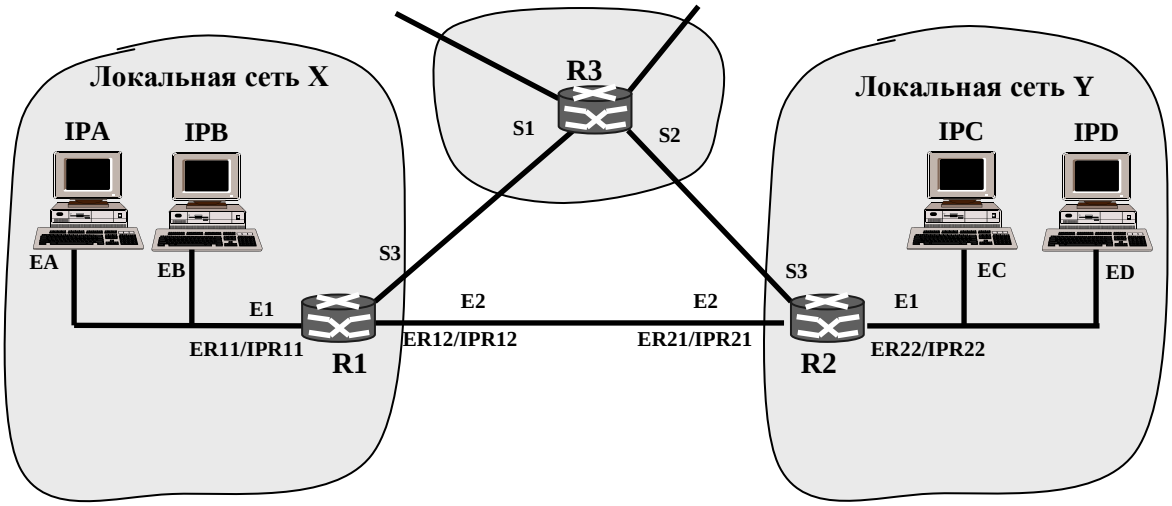

Рис. 2.9. Фрагмент IP-сети

Табл. 2.6 – Талица маршрутов хоста А

IP-адрес (IP-address)	Маска сети (Subnet Mask)	Адрес шлюза (Default Gateway)	Метрика
IP-A	255.255.192.0	IPR11	1

Из таблицы следует, что хост А подсоединен к сети X, номер которой определяется 18 первыми битами IP-A. Все пакеты, IP-адрес которых не совпал с адресом сети X, следует направлять через сеть на IP-адрес маршрутизатора IPR11.

В данном случае IP-адрес назначения совпадает с адресом собственной сети. Для перемещения дейтаграммы в локальной сети X модулю IP необходимо знать MAC-адрес хоста B. Для этого он обращается к своему модулю ARP, который хранит таблицы соответствия IP и MAC адресов. Допустим, что такой записи нет, поэтому модуль ARP отправляет в сеть X широковещательный запрос, содержащий собственный MAC-адрес EA и IP-адрес узла назначения IPB. Хост B, приняв запрос, в ответ высылает свой MAC-адрес EB модулю ARP хоста A, который 1)в своей таблице соответствия адресов создает запись “IPB соответствует EB” для будущего использования и 2)отдает принятый MAC адрес хоста B своему модулю IP. Теперь модуль IP хоста A отправляет дейтаграмму хосту B с MAC-адресом EB. В этом случае IP-адрес хоста B потребовался для определения его MAC-адреса.

Теперь предположим, что от приложения хоста A в модуль IP поступает дейтаграмма с IP-адресом назначения IPC. Допустим, что хост A ранее выполнял пересылку пакетов в маршрутизатор и ему известен его MAC-адрес ER11. Модуль IP хоста A формирует кадр Ethernet с MAC-адресом назначения ER11 и пересылает по локальной сети дейтаграмму маршрутизатору R1.

Таблица маршрутов маршрутизатора R1 может иметь различный вид, в зависимости от производителя и типа используемых интерфейсов. Если интерфейс использует протоколы Ethernet, то в таблице маршрутизации используется IP-адрес следующего перехода, а в случае использования протокола PPP – собственный выходной интерфейс маршрутизатора. Сама таблица маршрутизации может иметь следующий вид (табл. 2.7).

Табл. 2.7 - Таблица маршрутов маршрутизатора R1

Пункт назначения (Destination)	Маска сети (Subnet Mask)	Пункт пересылки (Next Hop)	Интерфейс (Interface)	Метрика (Metric)
Network X	255.255.192.0	IPR11	e1	1
Network Y	255.255.0.0	IPR21	e2	2
Network Y	255.255.0.0	0.0.0.0	s3	4
Default		0.0.0.0	s3	9

Модуль IP маршрутизатора руководствуется следующими правилами:

1. Никогда не меняет IP-адрес источника и IP-адрес назначения (исключение составляют пакеты, имеющие опциональные поля).
2. Никогда не направляет пакет на тот порт, от которого он поступил.
3. Использует только три нижних уровня модели OSI.
4. При наличии нескольких альтернативных маршрутов выбирает маршрут с меньшей метрикой (стоимостью).

Модуль IP маршрутизатора R1 анализирует IP-адрес назначения и обнаруживает, что он принадлежит сети Y. В таблице маршрутизации имеется две записи (вторая и третья строки), совпадающие с IP-адресом назначения, причем первый маршрут через порт 2 (IP-адрес порта IPR21) имеет меньшую метрику (Metric=2). Таблица маршрутов не хранит сведений о MAC-адресах, но они имеются в другой – ARP-таблице (не приведена), где имеется запись о том, что IP-адресу IPR21 соответствует MAC-адрес ER21. Модуль IP маршрутизатора R1 по протоколу Ethernet направляет пакет (IP-адрес назначения = IPC, MAC-адрес назначения = ER21, IP-адрес источника = IPA, MAC-адрес источника = ER12) через свой порт 2 в порт 1 маршрутизатора R2, где модуль канального уровня порта 1 принимает пакет и в виде дейтаграммы передает его в свой модуль IP. Интерфейс между маршрутизаторами R1 и R2 интерпретируется как вырожденная сеть Ethernet, содержащая два узла, IP-адреса IPR12 и IPR21 не должны совпадать с диапазонами IP-адресов сетей X и Y и обычно выбираются из диапазона частных адресов. Маршрутизаторы R1 и R3 между собой используют протокол PPP, который не требует выделения ни IP-адреса, ни MAC-адреса, они заменяются именем собственного выходного интерфейса (порта).

Маршрутизатор R2 имеет три интерфейса (порта) и примерно следующую таблицу маршрутов (табл. 2.8).

Табл. 2.8 - Таблица маршрутов маршрутизатора R2

Пункт назначения (Destination)	Маска сети (Subnet Mask)	Пункт пересылки (Next Hop)	Интерфейс (Interface)	Метрика (Metric)
Network Y	255.255.0.0	IPR22	e1	1
Network X	255.255.192.0	IPR12	e2	3
Network X	255.255.192.0	0.0.0.0	s3	6
Default		0.0.0.0	s3	8

Модуль IP маршрутизатора R2 анализирует IP-адрес назначения IPC и обнаруживает, что он совпадает с адресом собственной сети Y (Metric=1). Маршрутизатор по локальной сети Y через порт 2 передает пакет узлу C , используя те же процедуры, которые были рассмотрены при передаче пакета в локальной сети от узла A к узлу B.

ВОПРОСЫ К РАЗДЕЛУ 2.2

1. Сколько уровней содержит модель OSI?

Ответ. Семь.

2. Куда перемещаются пакеты со второго уровня при приеме?

Ответ. На третий уровень.

3. Куда перемещаются пакеты со второго уровня при передаче?

Ответ. На первый уровень.

4. Чем определяется количество уровней в узле?

Ответ. Сложностью задачи.

5. Чем определяются средства сетевого уровня для обработки пакетов, поступающих с канального уровня?

Ответ. Типом протокола: 0x0806 – для ARP и 0x0800 – для IP.

6. Чем определяются средства транспортного уровня для обработки пакетов, поступающих с сетевого уровня?

Ответ. Номером протокола: 0x0006 – для TCP и 0x0011 – для UDP.

7. Чем определяется тип сетевого приложения для обработки дейтаграмм?

Ответ. Номером порта.

8. Приведите примеры номеров портов общесетевых приложений.

Ответ: Порт 80 – HTTP, порт 23 – TELNET, порт 53 – DNS.

9. Назовите состав и назначение семейства протоколов PPP.

Ответ. LCP – для согласования параметров обмена пакетами, CHAP – для установления легитимности пользователя, IPCP – для присвоения IP-адреса.

10. Обеспечивает ли протокол PPP обнаружение ошибок и упорядоченную доставку пакетов?

Ответ. Обнаружение ошибок – да, упорядоченную доставку – нет, это обеспечивает протокол TCP.

11. Где хранятся данные для аутентификации пользователя?

Ответ. В Интернет-карте и на сервере AAA.

12. Можно ли до установления соединения с сервером NAS заранее определить IP-адрес пользователя?

Ответ: Нет. После успешной аутентификации NAS выдает свободный IP-адрес из диапазона (пула) отведенных адресов.

13. Какие способы используется для учета стоимости соединений в Интернет?

Ответ: Обычно взимается абонентская плата или плата за объем принятой информации.

14. Найти 87651234 mod 19.

Ответ. 87651234 mod 19 = 16.

15. Найти результат сложения чисел 123 и 456, используя операцию логического сложения “И”.

Ответ. 110110011_2 = 435.

16. Какую информацию содержит ARP-таблица?

Ответ. Соответствие между IP и MAC-адресами.

17. В каком случае устройство инициирует ARP-запрос?

Ответ. Устройство имеет IP-адрес назначения, но в своей ARP-таблице не может найти соответствующий ему MAC-адрес.

18. Какие уровни стека протоколов использует маршрутизатор?

Ответ. С 1 по 3.

19. Может ли в публичной сети существовать IP-адрес назначения 10.2.3.4?

Ответ. Нет, адрес сети 10.0.0.0 используется только в частных сетях.

20. Может ли в таблице маршрутизации существовать IP-адрес интерфейса 10.2.3.4 для пересылки пакета между соседними маршрутизаторами?

Ответ. Да.

21. Какая длина MAC-адреса в версии IPv4?

Ответ. 6 байт.

22. Какая длина IP-адреса в версии IPv4?

Ответ. 4 байта.

23. Что означает префикс IP-адреса в версии IPv4?

Ответ. Количество лидирующих бит, относящиеся к адресу сети/подсети.

2.3. Базовые протоколы Интернет

2.3.1. Физический уровень

Назначение физического уровня – надежная передача бит между отдельными узлами. Для того чтобы приемник точно знал, в какой момент времени нужно считать принимаемый бит, используются различные системы синхронизации. Внутри узла близко расположенные модули обычно обмениваются информацией двоичным биполярным потенциальным кодом NRZ (Non Return to Zero), используя тактовые частоты от одного и того же задающего генератора (ЗГ). В коде NRZ двоичной “1” соответствует напряжение не менее 2,4 вольта, а двоичному нулю – не более 0,4 вольта.

Код NRZ содержит постоянную составляющую и плохо приспособлен для передачи по длинным металлическим линиям с большой емкостью между проводами. Вдобавок приведенная схема плохо приспособлена для передачи на сколь либо длинное расстояние из-за необходимости прокладки отдельных трасс раздачи тактовой частоты от ЗГ к каждому узлу. Поэтому исходную передаваемую информацию следует преобразовать так, чтобы устранить постоянную составляющую, и из нее можно было бы извлечь информацию о тактовой частоте. Код NRZ мало приспособлен для этой цели в случае длительной передачи последовательных “1” или “0”. В зависимости от способа выделения тактовой частоты передачи различают две системы синхронизации: асинхронную и синхронную.

В асинхронной системе передача информации производится байтами, передатчик и приемник работают на своих тактовых частотах близких по номиналу. Начало передачи каждого байта передатчик предваряет специальным сигналом (битом) “старт”, по которому приемник настраивает свой ЗГ на моменты считывания информации. Окончание передачи байта завершается сигналом “стоп” (один/два байта), после которого передатчик выдерживает определенную паузу для подготовки приемника на прием следующего байта. За время передачи байта ЗГ приемника не успевает существенно разойтись по частоте с ЗГ передатчика, а при приеме следующего байта происходит его новая настройка.

В синхронной системе передача ведется непрерывно, без пауз между байтами. Для корректного приема передаваемый сигнал кодируют так, чтобы из него можно было извлечь информацию о тактовой частоте. Такие коды называются самосинхронизирующимися, а устройство их осуществляющее – преобразователем линейных кодов. В локальных сетях Ethernet для скоростей 10 Мбит/с широко применяется линейный код ***манчестер***. Код манчестер не содержит постоянной составляющей. Правило преобразования исходного кода NRZ в манчестерский: каждый исходный бит передается двумя битами противоположной полярности, причем первый имеет инверсное значение, второй – истинное (рис. 2.10).

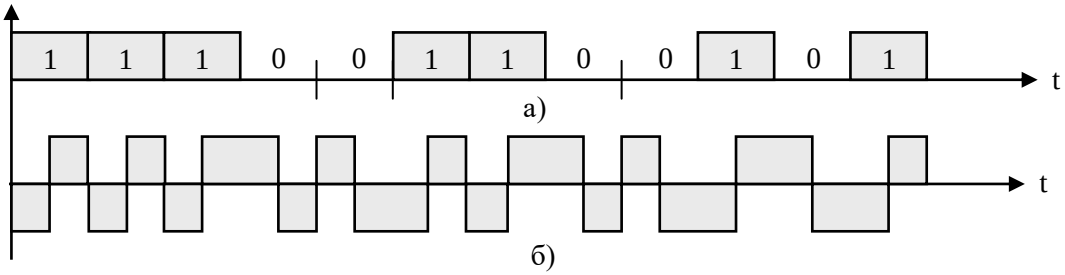

Рис. 2.10. Преобразование в манчестерский код
а) код NRZ б) манчестерский код

В технологии Fast Ethernet (100 Мбит/с) для передачи по металлическим линиям используются избыточный самосинхронизирующийся код 4B/5B. Правило преобразования исходного кода в линейный состоит в замене 4 бит исходной комбинации (16 комбинаций) на 5 бит (32 комбинации). Линейные коды подобраны таким образом, чтобы в линию передавалось бы максимальное количество “1”, независимо от исходного кода. Комбинаций линейных кодов в два раза больше комбинаций исходного кода, поэтому половина линейных кодов является запрещенной. Буква B в названии линейного кода означает, что используется два состояния линейного сигнала. Такая замена приводит к увеличению тактовой частоты передачи со 100 Мбит/с до 125 Мбит/с, и для ее понижения используется линейный код MLT-3. Правило преобразования исходного кода NRZ в MLT-3: “1” – передаются дебитами противоположной полярности, причем первая половина повторяет предыдущую полярность, “0” – сохранением полярности (рис. 2.11).

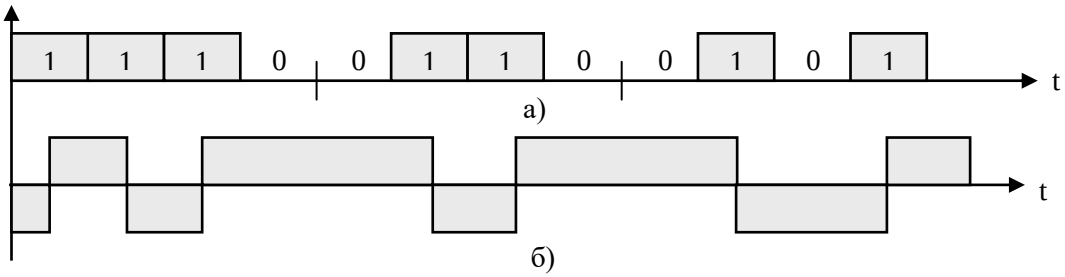

Рис. 2.11. Преобразование в MLT-3
а) код NRZ б) код MLT-3

2.3.2. Канальный уровень

На канальном уровне осуществляется коммутация пакетов в локальной сети. Существует много технологий канальных уровней – Ethernet, Token Ring, FDDI и другие. Каждая из технологий рассчитана на определенную топологию: Ethernet – на подключение узлов к общей шине, а Token Ring и FDDI имеют кольцевую топологию

Технология ***Ethernet*** (***RFC 894*** [42]) – это самая распространенная технология локальных сетей. В зависимости от типа физической среды имеются различные стандарты – 10Base-5, 10Base-2, 10Base-T, 10Base-F. Число 10 обозначает скорость передачи 10 Мбит/с, Base – передачу на одной базовой частоте 10 МГц, последний символ обозначает тип кабеля (5 – коаксиальный кабель диаметром 0.5 дюйма, 2 – коаксиальный кабель диаметром 0.25 дюйма, T – неэкранированная витая пара, F – оптический кабель).

Во всех вариантах на физическом уровне используется манчестерский код. В сетях Ethernet используется множественный доступ с контролем несущей и обнаружением коллизий (Carrier Sense Multiply Access with Collision Detection, ***CSMA/CD***). Все компьютеры сети имеют доступ к общей шине через встроенный в

каждый компьютер сетевой адаптер, используя полудуплексный режим передачи. Схема подключения компьютеров по коаксиальному кабелю приведена на рис. 2.12.

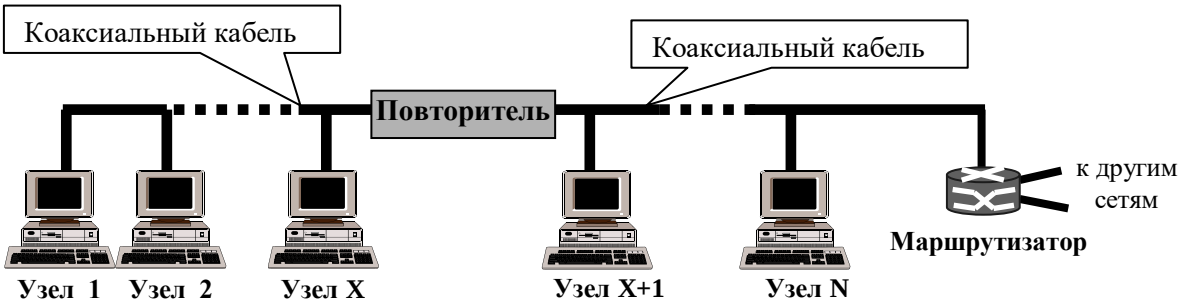

Рис. 2.12. Сеть Ethernet на коаксиальном кабеле (стандарты 10Base-5 / 10Base-2)

Перед началом передачи узел должен убедиться в отсутствии занятия несущей среды, признаком которого является отсутствие на ней несущей частоты. Если среда свободна, то узел имеет право начать передачу кадра определенного формата. Предположим, что узлу 2 требуется передать кадр узлу N. Обнаружив, что среда свободна, узел 2 начинает передачу кадра (рис. 2.13), которая предваряется *преамбулой (preamble),* состоящей из 7 байт вида 10101010, и байта *начала кадра (Start of Frame Delimiter, SFD)* вида 10101011. Эти комбинации нужны приемнику для вхождения в побитовый и кадровый синхронизм с передатчиком. Кадр заканчивается полем контрольной суммы (FCS) длиной 4 байта (на рис. 2.13 не показано). Сигналы передатчика распространяются по кабелю в обе стороны, и все узлы распознают начало передачи кадра. Только узел N опознает свой собственный адрес (MAC-адрес назначения) в начале кадра и записывает его содержимое в свой буфер для обработки. Из принятого кадра определяется адрес источника (MAC-адрес источника), которому следует выслать кадр-ответ.

Получатель пакета на 3-м уровне определяется в соответствии с полем *Тип протокола (Protocol Type)*: значение 0x0800 – адрес модуля IP, 0x0806 – адрес модуля ARP. Минимальное и максимальное значения длины поля для протоколов верхних уровней – 46 и 1500 байт, соответственно. Порядок передачи бит кадра: слева направо/снизу вверх (рис. 2.13), цифрами обозначены длины полей кадра в байтах.

Рис. 2.13. Местоположение в стеке протоколов и форматы физического и канального уровней сети Ethernet

Любой узел при наличии кадра к передаче и занятой среде вынужден ждать ее освобождения. Признаком окончания передачи является пропадание несущей частоты. После окончания передачи кадра все узлы должны выдержать технологическую паузу 9.6 мкс, чтобы привести сетевые адаптеры в исходное состояние и предотвратить

повторный захват среды одним и тем же узлом. Иногда возникают ситуации, когда один узел уже начал передачу, но другой узел еще не успел это обнаружить и также начинает передачу своего кадра. Такая ситуация захвата свободной среды более, чем одним узлом, называется *коллизией*. Механизм разрешения коллизии состоит в следующем.

Узел, ведущий передачу кадра, одновременно наблюдает за приемом. Если уровень принимаемого сигнала не превышает порогового значения, то узел продолжает передачу, если же превышает, то узел прекращает передачу кадра и посылает в сеть специальную 32-битную jam-комбинацию с нерегламентированной последовательностью, просто приводящей к повышению уровня сигнала в локальной сети, из-за увеличения амплитуды импульсов манчестерского кода суммарного сигнала. После этого узел, обнаруживший коллизию, делает случайную паузу и затем снова может повторить попытку передачи кадра. Число повторных попыток не может превысить 16. Если же и после 16-й попытки кадр вызвал коллизию, то он отбрасывается. При большом количестве узлов вероятность коллизии возрастает, и пропускная способность сети Ethernet падает, т.к. сеть все большее время занята обработками коллизий и отбрасыванием кадров.

Для увеличения длины общей сети, состоящей из различных сегментов кабеля, используются повторители. *Повторитель* является устройством 1-го уровня, он принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их на другом сегменте кабеля, увеличивая мощность и улучшая форму импульсов. Применение повторителя вносит дополнительную задержку и ухудшает распознавание коллизии, поэтому их количество в сети Ethernet не должно превышать 4, при этом максимальная длина одного сегмента должна быть не более 500 метров, а диаметр всей сети – не более 2500 метров.

Наряду с очевидной простотой стандарты 10Base-5 и 10Base-2 обладают существенным недостатком: трудоемкостью локализации неисправного сетевого адаптера для отсоединения поврежденного узла.

Сеть Ethernet стандарта 10Base-T предусматривает сохранение всех протоколов канального уровня стандартов 10Base-5 и 10Base-2 и требует замены только сетевого адаптера в каждом узле. Соединение узлов между собой осуществляется через центральное устройство – концентратор, что значительно устраняет недостатки предыдущих стандартов.

Концентратор (Hub) является устройством 1-го уровня и осуществляет функции повторителя на всех отрезках витых пар между концентратором и узлом, за исключением того порта, с которого поступает сигнал. Каждый порт имеет приемник (R) и передатчик (T). Кроме того, концентратор сам обнаруживает коллизию и посылает jam-последовательность на все свои выходы. Типовая емкость концентратора – от 8 до 72 портов.

Концентраторы можно соединять друг с другом с помощью тех же портов, которые используются для подключения узлов. Стандарт разрешает соединять концентраторы только в древовидные структуры, любые петли между портами концентратора запрещены. Для надежного распознавания коллизии между двумя любыми узлами должно быть не больше 4 концентраторов, при этом максимальная длина между концентраторами должна быть не более 100 метров, а диаметр всей сети – не более 500 метров.

В стандарте 10Base-F в качестве передающей среды используется оптоволоконный кабель. Он предусматривает сохранение всех протоколов канального уровня предыдущих стандартов и требует замены только сетевого адаптера.

Технология ***Fast Ethernet*** позволяет передавать кадры со скоростью 100 Мбит/с, аппаратно отличается от технологии ***Ethernet*** только на физическом уровне. В качестве

физической среды может использоваться только витая пара или оптоволоконный кабель; способ кодирования линейного сигнала – код 4B/5B или 8B/6T; максимальный диаметр сети – 200 метров. Последние модели концентраторов сами определяют скорость передачи подключенных узлов и автоматически подстраиваются под соответствующую скорость.

Построение локальных сетей большой емкости только с помощью концентраторов приводит к возрастанию числа коллизий и снижению пропускной способности сети. Поэтому концентраторы используются для построения небольших фрагментов сетей, которые затем объединяются с помощью мостов и коммутаторов.

Мост (Bridge) является устройством 2-го уровня, он также соединяет два сегмента сети (рис. 2.14) но, в отличие от повторителя, снабжен определенной логикой. Порт моста записывает все кадры, поступающие от узлов одного сегмента, в буферную память.

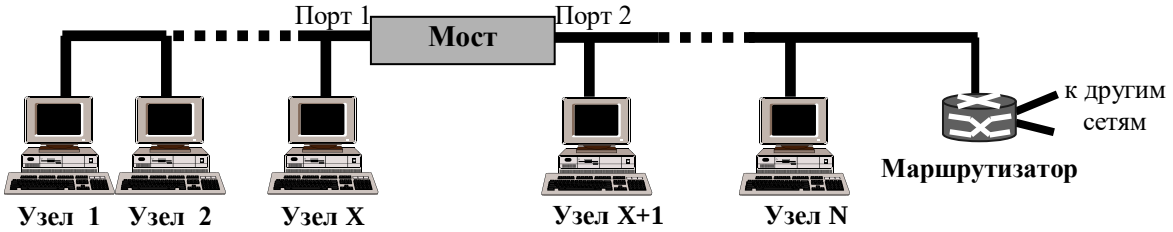

Рис. 2.14. Сеть Ethernet с использованием моста

В исходном состоянии, когда состав сети неизвестен, мост ретранслирует буферизированные кадры, поступающие с одного порта, на другой порт по алгоритму CSMA/CD. Одновременно мост создает таблицу принадлежности MAC-адреса каждого поступающего пакета конкретному порту. По истечении некоторого времени мост составит следующую таблицу (табл. 2.9).

Табл.2.9 – Таблица маршрутов моста

Порт 1	Порт 2
MAC-адрес узла 1	MAC-адрес узла X+1
MAC-адрес узла 2	. . .
. . .	MAC-адрес узла N
MAC-адрес узла X	MAC-адрес маршрутизатора

Если теперь мост получает кадр от узла 1 направленный к узлу N (рис. 2.14), то он по свой таблице определяет, что нужно ретранслировать кадр из буфера порта 1 в порт 2. Если же мост получает кадр от узла 2 направленный к узлу X, то он по своей таблице определяет, что узлы находятся в одном сегменте и стирает кадр в буферной памяти порта 1. Записи в таблице, произведенные мостом, являются *динамическими*. По истечении определенного короткого времени, если мост не принял ни одного кадра от какого-то узла, то такая запись стирается. Такая процедура ограниченного хранения записанных данных называется *кэшированием* (cache) и предохраняет таблицу от переполнения при удалении узла из сети или при перемещении узла из одного сегмента в другой. Записи, внесенные в таблицу администратором, называются *статическими* и не имеют срока жизни, они дают возможность администратору при необходимости принудительно подправлять работу моста.

Коммутатор (Switch) использует топологию вида звезда, является устройством 2-го уровня и функционально представляет собой многопортовый мост, к каждому порту которого может быть подключен отдельный хост, концентратор, сервер или маршрутизатор (рис. 2.15). Каждый порт коммутатора оснащен процессором обработки

пакетов (Пр), который может работать как в полудуплексном, так и в дуплексном режиме. При подключении к порту коммутатора отдельного узла (компьютера или маршрутизатора) порт коммутатора устанавливается в дуплексный режим, а в узле устанавливается сетевая карта с подавлением коллизий. За счет этого узел и коммутатор имеют возможность одновременной передачи и приема пакетов. При подключении к порту коммутатора концентратора порт коммутатора устанавливается в полудуплексный режим.

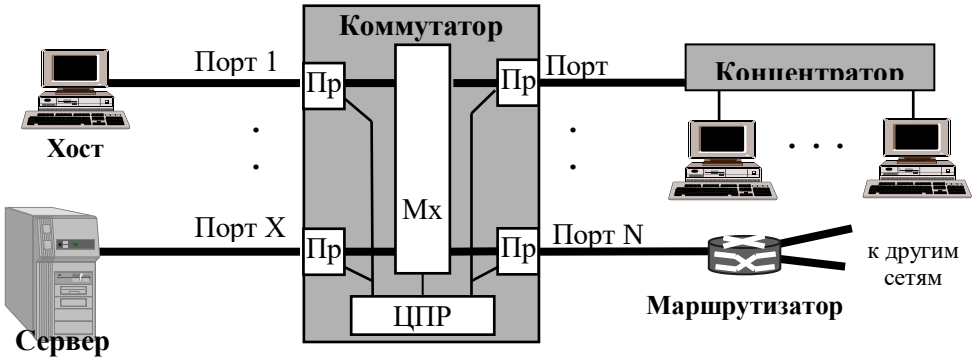

Рис. 2.15. Сеть Ethernet с использованием коммутатора

Для коммутации кадров между портами используется коммутационная матрица (Mx). Аналогично мосту каждый порт ведет адресную таблицу MAC-адресов (content-addressable memory, CAM), подключенных к нему устройств, и сообщает о ней центральному процессору (ЦПР). После приема начальных бит кадра входной процессор анализирует адрес назначения и пытается установить соединение через коммутационную матрицу, не дожидаясь прихода оставшихся бит кадра. Для этого он обращается к ЦПР с заявкой на установление пути в коммутационной матрице. ЦПР имеет адресную таблицу и может осуществить запрашиваемое соединение, если порт назначения свободен, т.е. не соединен с другим портом. Если же порт занят, то ЦПР в соединении отказывает, и кадр продолжает буферизироваться процессором входного порта до освобождения выходного порта. После того, как требуемый путь в коммутационной матрице установлен, по нему направляются кадры в выходной порт, где они повторно буферизируются на случай разной скорости коммутируемых портов. Процессор выходного порта по значению контрольной суммы (FCS) опционально может проверить целостность принятого кадра и начинает передавать по сегменту Ethernet принятый кадр. Из-за наличия множества портов коммутатор обладает существенно более высокой производительностью за счет параллельной обработки кадров. Если потоки данных между портами распределяются, не конфликтуя между собой, т.е. соединения вида входной порт-выходной порт образуют независимые пары, то коммутатор емкостью N портов может одновременно обслуживать N/2 полудуплексных соединений или N дуплексных соединений.

Технология ***Token Ring*** (***RFC 1513*** [53]) также использует разделяемую среду с топологией в виде кольца, но в отличие от Ethernet используется не случайный, а детерминированный доступ. Право на доступ передается от одного узла к другому в виде специального кадра, называемым *маркером* (Token). Каждый порт имеет приемник (R) и передатчик (T). Скорость передачи кадров по кольцу может быть 4 или 16 Мбит/с, смешение скоростей в одном кольце не допускается. В качестве физической среды может использоваться витая пара или оптоволоконный кабель. Максимальное число узлов в кольце – 260, максимальная длина кольца – 4000 метров.

В сети Token Ring любой узел, получив маркер от предыдущего узла, изымает его из кольца и, если поступил кадр от вышележащего уровня, начинает его передачу. Кадр имеет адрес назначения и адрес источника. Передаваемый кадр последовательно проходит по кольцу от одного узла к другому. Все узлы, собственный адрес которых не совпал с адресом назначения в кадре, ретранслируют кадр побитно, как повторители. Если приемный узел распознал свой адрес, то он копирует кадр в свой буфер и вставляет в ретранслируемый кадр признак подтверждения приема. Узел, инициировавший передачу кадра, получив подтверждающий кадр, изымает его из кольца и передает маркер следующему узлу. Каждый узел следит за временем удержания маркера, которое обычно равно 10 мс. За это время узел может передать несколько кадров.

При первоначальном включении все станции начинают процедуру выбора активного узла, критерием которого является узел с максимальным значением MAC-адреса. В дальнейшем этот узел каждые 3 секунды генерирует специальный кадр своего присутствия и восстанавливает маркер в случае его утери. В случае отключения какого-либо узла от сети или пропадания питания на узле релейные схемы этого порта замыкают накоротко его передатчик (T) с приемником (R), обеспечивая неразрывность кольца.

Технология FDDI (Fiber Distributed Data Interface, ***RFC 1390*** [52]) базируется на технологии Token Ring, т.е. использует разделяемую среду с топологией в виде кольца, но в качестве транспортной среды использует два оптоволоконных кабеля со скоростью передачи до 100 Мбит/с. Используется линейный код 4B/5B. За счет низкого затухания оптического сигнала максимальная длина кольца может достигать 100 километров.

В исправном состоянии сети кадры проходят через все узлы основного кольца, резервное кольцо при этом не используется. При повреждении основного кольца оптические переключатели (ОП) срабатывают так, что задействуется вторичное кольцо, при этом направление следования передаваемых кадров от узла к узлу не меняется. При наличии нескольких повреждений приводит к образованию нескольких независимых сетей.

2.3.3. Сетевой уровень

Основным протоколом сетевого уровня является *протокол Интернет*, которым пользуется IP-модуль для коммутации (перемещения) пакета с одного интерфейса на другой. Но коммутация в локальной сети происходит на основе MAC-адресов, поэтому IP-модуль пользуется таблицей соответствия вида IP-адрес – MAC-адрес, которую заполняет *протокол преобразования адресов*. Для нахождения оптимального маршрута IP-модуль использует таблицу маршрутизации, которую составляет *протокол маршрутизации*. О возникших проблемах маршрутизаторы извещают друг друга при помощи *протокола межсетевых управляющих сообщений***.** Рассылка одного и того же пакета множеству получателей производится *протоколом группового обслуживания*. Ниже приводится подробное описание упомянутых протоколов.

Протокол Интернет версии 4 (Internet Protocol, IPv4) согласно ***RFC 791*** [36] решает две базовые функции: адресацию и фрагментацию. Маршрутизацию пакета модуль IP осуществляет по адресу, имеющемуся в заголовке пакета. Там же имеется информация о способе фрагментации/сборке дейтаграмм, если максимальная длина сетевого пакета (Maximum Transfer Unit – MTU) отличается от значения MTU, принятого на адресуемом интерфейсе.

Формат IPv4 изображен на рис. 2.16. Порядок передачи бит кадра: слева направо/снизу вверх. Назначение полей пакета следующее.

<table>
<tr><td colspan="5">Данные (Data)</td></tr>
<tr><td colspan="5">Опции и заполнение (Options + Padding)</td></tr>
<tr><td colspan="5">Адрес назначения (Destination Address)</td></tr>
<tr><td colspan="5">Адрес источника (Source Address)</td></tr>
<tr><td colspan="2">Время жизни
(Time to Live)</td><td>Протокол
(Protocol)</td><td colspan="2">Контрольная сумма
(Header Checksum)</td></tr>
<tr><td colspan="3">Идентификатор (Identification)</td><td>Флаги
(Flags)</td><td>Смещение фрагмента
(Fragment Offset)</td></tr>
<tr><td>Версия
(Version)</td><td>Длина
(IHL)</td><td>Тип услуги
(Type of Service)</td><td colspan="2">Общая длина
(Total Length)</td></tr>
<tr><td colspan="5">2 уровень</td></tr>
<tr><td colspan="5">1 уровень</td></tr>
</table>

Рис. 2.16. Формат IPv4

Версия (Version) – определяет версию протокола, длина 4 бита, для версии 4 Version=4 (двоичный код 0100).

Длина заголовка (Internet Header Length – IHL) – определяет длину заголовка в 32-х битных словах, минимальная значение – 5 (20 байт), максимальное – 15 (60 байт).

Тип услуги (Type of Service – TOS) – длина 8 бит, задает приоритетность обслуживания пакета и критерий выбора маршрута следующим образом. Биты 0-2: Приоритет (Precedence) – определяют IP-приоритет пакета (111 – наивысший приоритет, 000 – низший), биты 3-6 указывают на предпочтение при выборе маршрута (код 1000 – низкая задержка; код 0100 – высокая пропускная способность; код 0010 – высокая надежность, код 0001 – низкая стоимость, 0000 – обычное обслуживание). Бит 7 – резерв.

Общая длина пакета (Total Length) – определяет длину пакета (заголовок + данные) в байтах, длина 16 бит (максимальная длина пакета 65535 байт). Все оконечные устройства обязаны обрабатывать пакеты длиной до 576 байт. Использование пакетов большей длины допустимо при способности их обработки всеми промежуточными узлами.

Идентификатор (Identification) – указывает на принадлежность пакета к определенной дейтаграмме, длина 16 бит. Все фрагменты (пакеты) одной дейтаграммы должны иметь одно и то же значение.

Флаги (Flags) – содержат признаки фрагментации, длина 3 бита. Первый бит – резервный, устанавливается в 0, второй бит DF содержит указание маршрутизаторам о возможности (DF=0) или невозможности (DF=1) фрагментации пакета, третий бит MF указывает на последний (MF=0) или промежуточный (MF=1) пакет дейтаграммы. Для не фрагментированных дейтаграмм MF=0.

Смещение фрагмента (Fragment Offset - FO) – указывает на положение пакета относительно начала дейтаграммы, длина 13 бит (максимальное число фрагментов одной дейтаграммы 8192). Смещение кратно 8 байт (64 бит), первый пакет дейтаграммы имеет значение этого поля равное нулю. Для не фрагментированных дейтаграмм FO=0.

Время жизни (Time to Live – TTL) – определяет предельный срок перемещения пакета по сети, длина 1 байт. Устанавливается на передаче, каждый маршрутизатор вычитает из текущего значения единицу, при значении TTL=0 пакет уничтожается.

Протокол (Protocol – Prot) – указывает на протокол, которому следует доставить (или от которого поступила) информация, находящаяся в поле данных для дальнейшей обработки: ICMP (Prot=1), TCP (Prot=6), UDP (Prot=17), SIP (Prot=41), …

Контрольная сумма заголовка (Header Checksum) – используется для обнаружения ошибок принятого пакета, длина 16 бит. Вычисляется каждым маршрутизатором при передаче пакета заново из-за изменения TTL. Формируется так, чтобы сумма всех 16-разрядных слов заголовка, включая контрольную сумму, давала бы 0xFFFF. Никаких других процедур по повышению достоверности передачи (обнаружение ошибок в поле данных, подтверждения о правильном приеме, повторная передача, управление потоком) не используется.

Адрес источника (Source Address) – определяет сетевой адрес (IP-адрес) отправителя пакета, длина 32 бита. Структура сетевых адресов приведена в 1.5.2.

Адрес назначения (Destination Address) – определяет сетевой адрес получателя, длина 32 бита. Структура сетевых адресов приведена в 1.5.2.

Опции (Options) – это поле может отсутствовать, имеет переменную длину. Обычно используется при отладке сети. В первом байте поля содержится тип опции (Opt Code), во втором – длина содержимого, в последующих байтах – содержимое. Поле с типом опции “0” является закрывающим, а с типом “1” – пустым.

Заполнение (Padding) – содержит заполнение пакета “нулями” так, чтобы общая длина пакета была кратна 32 бит.

В поле *данных (Data)* размещаются информация, которую должны обрабатывать протоколы, указанные в поле *Протокол (Protocol – Prot)*: ICMP (Prot=1), TCP (Prot=6), UDP (Prot=17), …

Каждый интерфейс маршрутизатора имеет ограничение по передаче максимальной длины пакета MTU. Когда происходит пересылка пакета с одного интерфейса на другой, имеющий меньшее значение MTU, происходит фрагментация длинного пакета на пакеты меньшей длины.

Опциональные поля используются администратором сети для настройки и тестирования маршрутов. Обычно используются три опции: *свободная маршрутизация, заданная маршрутизация, запись полного маршрута*.

Для установления соединения в Интернет абонент-инициатор набирает известный ему сетевой адрес (IP-адрес) своего партнера, однако, для коммутации в локальной сети от одного узла к другому узлу используются локальные адреса (MAC-адреса). Поэтому передающий IP-модуль устанавливает соответствие между известным сетевым и неизвестным локальным адресом приемного узла либо на основании заранее составленной таблицы, либо обращается к модулю ARP, который формирует широковещательный запрос локального адреса, используя ***протокол преобразования адресов (Address Resolution Protocol, ARP).*** Формат ARP (***RFC 826*** [40]) приведен на рис. 2.17. Порядок передачи бит кадра: слева направо/снизу вверх, верхними цифрами обозначены длина полей кадра в байтах.

Рис. 2.17. Формат ARP

В поле третьего уровня *тип интерфейса* (*Hardware Type, Htype*, длина 2 байта) помещается код технологии локальной сети (0x0001 – технология Ethernet), в поле *Протокол* (длина 2 байта) – адрес инициатора запроса (0x0800 – код модуля IP), в поле *длина MAC-адреса* (*Hardware Address Length, Hlen*) и *длина IP-адреса* (по 1 байту) – длины MAC- и IP- адресов, соответственно. Поле *операция* (длина 2 байта) принимает значение “Запрос” (код 0x0001) или “Ответ” (код 0x0002). В поле *IP-адреса источника* и *MAC-адреса источника* размещается свой IP-адрес (4 байта) и свой MAC-адрес (6 байт). В поле *IP-адреса назначения* помещается IP-адрес назначения (4 байта), а поле *MAC-адрес назначения* заполняет “нулями”.

Модуль ARP формирует поля второго уровня, подставляя в поле *MAC-адреса источника* собственный адрес, в поле *MAC-адрес назначения* – широковещательный запрос, в поле *Тип протокола* – код модуля ARP (0x0806) и рассылает запрос всем устройствам, находящимся в локальной сети. Только то устройство, которое опознает в поле *IP-адреса назначения* свой IP-адрес, отвечает на запрос и заполняет пустое поле *MAC-адрес назначения* своим значением.

Цель задачи маршрутизации – определение наилучшего пути от источника до получателя, при этом предполагается, что в общем случае от источника до получателя имеется несколько путей и пакет проходит через несколько сетей (маршрутизаторов). Адресация сетей может не совпадать с классической адресацией в IP-сети из-за использования масок. Для установления соединений в сети каждый маршрутизатор (шлюз) должен иметь информацию о путях достижения заданного узла назначения. Каждый маршрутизатор хранит эту информацию в виде таблицы маршрутов, которая может заполняться различным образом. Таблица может заполняться вручную администратором сети, что требует от него оценки целесообразности использования каждого пути и постоянного слежения за изменением структуры сети и изменением статуса прописанных путей, т.к. ситуация на сети регулярно изменяется из-за повреждения интерфейсов и их восстановления. С возрастанием емкости сети объем таблиц, и трудоемкость их заполнения становится слишком большими. Для устранения этого, структура Internet представляется как совокупность автономных систем (Autonomous System, AS), подсоединенных к общей глобальной сети (Core Backbone Network). В автономной системе используется ***единая*** политика маршрутизации.

Автономная система включает в себя несколько региональных сетей (зон), которые объединяет магистральная сеть. Магистральная сеть является отдельной зоной автономной системы и частью глобальной сети. Все маршрутизаторы одной зоны хранят идентичную базу данных маршрутов достижения других зон и автономных систем. Каждая зона имеет свою топологическую базу данных. Внутри автономной системы маршрутизаторы делятся на классы, различают внутренние маршрутизаторы (Internal routers), граничные маршрутизаторы зоны (Area border routers, ABR), граничные маршрутизаторы автономной системы (AS border routers, ASBR). Каждый интерфейс имеет свою метрику.

Зоны бывают транзитные и тупиковые. Транзитные зоны имеют более одного граничного маршрутизатора зоны, тупиковые – один. Между собой сетевые маршрутизаторы связываются либо через широковещательные сети, либо непосредственно, либо по виртуальным каналам. Широковещательные сети характеризуется наличием хотя бы одной сети, к которой подсоединены все маршрутизаторы зоны. В зависимости от места нахождения маршрутизатора – внутри автономной системы или на ее границе – различают внутренние (Interior Gateways) и внешние (Exterior Gateways) шлюзы. Каждый из них использует свои протоколы для рассылки маршрутной информации. Внутренний ***протокол маршрутизации выбора кратчайшего пути первым (Open Shortest Path First, OSPF)*** согласно ***RFC 2328*** [64]

является составной частью модуля IP и используется для рассылки различного типа маршрутной информации внутри автономной системы. Эта рассылка между маршрутизаторами происходит в виде ***сообщений о состоянии связей (Link State Advertisement, LSA)***. Совокупность всех типов имеющейся маршрутной информации образует базу данных маршрутизатора.

Изначально администратор сети вручную создает минимальную таблицу связей для каждого маршрутизатора, учитывая только непосредственно подсоединенные соседние маршрутизаторы и сети. В описание связи входят класс маршрутизатора (внутренний или граничный), IP-адрес узла, маска, метрика. Метрика характеризует цену использования выбранного исходящего пути: пути с меньшей метрикой более предпочтительны. При начальном включении маршрутизатор определяет, какие интерфейсы у него активны. После этого маршрутизатор сам начинает обнаруживать соседние маршрутизаторы путем рассылки протокольного сообщения Hello и приема от них таких же сообщений. Сообщение Hello также используется для выбора представителя зоны (Designated Router, DR) в широковещательных сетях.

На первом этапе начинают сопрягаться между собой внутренние маршрутизаторы рассылкой сообщений Hello. В этом сообщении каждый маршрутизатор подставляет свой IP-адрес и список IP-адресов соседних маршрутизаторов, от которых принято сообщение Hello. Это же сообщение используется для выбора представителя зоны (Designated Router, DR). Сопряженные маршрутизаторы начинают обмен маршрутной информацией в виде сообщений о состоянии связей (Link State Advertisement, LSA) и формировать свою базу данных. Получив новое сообщение, маршрутизатор реконструирует топологическую древовидную базу данных связей с другими узлами, считая себя корнем. Из множества альтернативных путей внутри зоны вычисляется путь с наименьшей метрикой до сопряженного маршрутизатора. Совокупность таких маршрутов ко всем известным внутренним маршрутизаторам образует таблицу наикратчайших путей внутри зоны. В дальнейшем маршрутизаторы продолжают периодически высылать сообщение Hello для обозначения своего присутствия в сети и обмениваться маршрутной информацией. Сообщения о состоянии связей рассылается и в случае изменения какой-либо связи. Такое взаимодействие между внутренними маршрутизаторами позволяет достаточно быстро обнаружить неисправный маршрутизатор и реконструировать свою базу данных.

Граничные маршрутизаторы зоны (ABR) по определению соединены с определенной зоной и магистральной сетью и имеют отдельную базу данных для каждой из зон (внутренней и магистральной). По совместительству один из граничных маршрутизаторов всегда является представителем зоны. *На втором этапе* они суммируют маршрутную информацию, полученную от своей зоны, для ее рассылки в другие ABR через магистральную сеть автономной системы и принимать такую же информацию от других ABR. В результате обмена они получают полную топологическую информацию о магистральной сети и о сетях в других зонах. На основании этой информации граничные маршрутизаторы зоны вычисляют путь с наименьшей метрикой до каждого другого ABR и рассылают таблицу связей с другими ABR всем внутренним маршрутизаторам зоны, что дает последним возможность вычисления наикратчайшего пути к другим зонам.

Граничные маршрутизаторы автономной системы (ASBR) производят обмен маршрутной информацией с граничными маршрутизаторами других автономных систем. *На третьем этапе* они распространяют и получают маршрутную информацию о внешних сетях всем граничным маршрутизаторам зоны ABR через автономную систему. Граничные маршрутизаторы зоны ABR транслируют ее всем внутренним маршрутизаторам своей зоны, что дает последним возможность вычисления

наикратчайшего пути к сетям других автономных систем. Исключение составляют тупиковые сети, в которых администратором прописываются постоянные маршрутные пути.

OSPF расположен над IP, используя номер протокола 89 (Protocol=89). OSPF не предусматривает фрагментацию длинных сообщений, поэтому длинные сообщения разбиваются модулем IP на несколько коротких сообщений без потери функциональности. OSPF представляет семейство сообщений для передачи информации различного типа и состоит из заголовка и содержимого. Нормально типы сообщений используются в следующем порядке.

При включении маршрутизатора он начинает высылать сообщения Hello для обнаружения соседних маршрутизаторов и установления взаимосвязи с ними по IP-адресу AllSPFRouters=224.0.0.5. В дальнейшем каждый маршрутизатор периодически продолжает высылать эти сообщения для подтверждения своего присутствия в сети. Маршрутизатор начинает рассылку списка известных ему связей соседним маршрутизаторам, используя сообщение Database Description.

Получив список, каждый маршрутизатор сличает его с имеющимися у него данными и при отсутствии описания какой-либо связи запрашивает ее в сообщении Link State Request.

Получив запрос о связи, соседний маршрутизатор высылает запрошенные данные в сообщении Link State Update. Процесс обмена данными о связях называется ***синхронизацией базы данных***. После полной синхронизации (идентичности) базы данных соседние маршрутизаторы становятся ***сопряженными.***

В случае приема нового или одного и того же сообщения дважды маршрутизатор высылает источнику сообщение Link State Acknowledgment.

Соединения в IP-сети устанавливают маршрутизаторы, которые при возникновении каких-либо проблем установления соединения должны извещать друг друга об ошибочных ситуациях. Для этих случаев используется ***протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP),*** являющийся составной частью IP и имеющийся в каждом модуле IP. ICMP-сообщения высылаются в различных случаях: при невозможности установления требуемого соединения, при отсутствии необходимого объема памяти для буферизации кадров, для извещения узла о наличии более короткого маршрута и т.д. Сообщения передаются модулем IP, который не является абсолютно надежным, ни какие сообщения о проблемах самого ICMP не посылаются.

ICMP (***RFC 792*** [37]) расположен в модуле IP, используя номер протокола 1 (Prot=1). Местоположение и наименование полей ICMP изображено на рис. 2.18. Порядок передачи бит кадра: слева направо/снизу вверх. Назначение полей пакета следующее.

Тип ошибки (Type) – в это поле помещается код типа ошибки, с кратким описанием причины ошибки в поле *Код ошибки (Code)*.

Контрольная сумма сообщения ICMP (Checksum) – используется для обнаружения ошибок при передаче ICMP-пакета, длина 16 бит. Сумма всех 2-байтных слов сообщения от поля *Type* совместно с контрольной суммой дает результат 0xFFFF.

Сообщения ICMP (ICMP Data) – здесь размещается дополнительная информация с пояснением или указанием ошибки.

Данные (Data) – в это поле обычно помещается заголовок принятого с ошибками IP-кадра и его первые 64 бита данных.

<table>
<tr><td colspan="6">Данные (Data)</td></tr>
<tr><td colspan="6">Сообщения ICMP (ICMP Data)</td></tr>
<tr><td colspan="2">Тип ошибки (Type)</td><td colspan="2">Код ошибки (Code)</td><td colspan="2">Контрольная сумма сообщения (Checksum)</td></tr>
<tr><td colspan="6">Адрес назначения (Destination Address)</td></tr>
<tr><td colspan="6">Адрес источника (Source Address)</td></tr>
<tr><td colspan="2">Время жизни
(Time to Live)</td><td colspan="2">Протокол
(Protocol = 1)</td><td colspan="2">Контрольная сумма заголовка
(Header Checksum)</td></tr>
<tr><td colspan="4">Идентификатор (Identification)</td><td>Флаги
Flags=0</td><td>Смещение фрагмента
(Fragment Offset=0)</td></tr>
<tr><td>Версия
(Version)</td><td>Длина
(IHL)</td><td colspan="2">Тип услуги
(Type of Service=0)</td><td colspan="2">Общая длина пакета (Total Length)</td></tr>
<tr><td colspan="6">2 уровень</td></tr>
<tr><td colspan="6">1 уровень</td></tr>
</table>

Рис. 2.18. Местоположение в стеке протоколов и формат ICMP

Совокупность полей *Type, Code, ICMP Data, Data* для некоторых типовых случаев приведена в табл. 2.10.

Табл. 2.10 – Совокупность некоторых полей ICMP

Тип (Type)	Коды сообщений (Code)	Данные (ICMP Data)	Примечания (Data)
3	Недоступность: 0 – сети, 1 – узла, 2 – протокола, 3 – порта, 4 – требуется фрагментация при установленном флаге “запрет фрагментации”, 5 – поврежден маршрут, …	Пусто	Заголовок принятого IP-кадра и его первые 64 бита данных
4	0 – не хватает памяти для буферизации пакета	Пусто	Заголовок принятого IP-кадра и его первые 64 бита данных
5	Указание о более коротком пути: 0 – к сети, 1 – к узлу, 2 – к сети из-за типа услуги, 3 – к узлу из-за типа услуги	IP-адрес подходящего шлюза	Заголовок принятого IP-кадра и его первые 64 бита данных
11	0 – истекло время жизни кадра, 1 – истекло время фрагментации кадра	Не используется	Заголовок принятого IP-кадра и его первые 64 бита данных
12	0 – на место ошибки указывает указатель (в байтах)	Указатель на ошибку в заголовке (например, 3 – поле общей длины пакета)	Заголовок принятого IP-кадра и его первые 64 бита данных

В Интернет предусмотрена возможность групповой рассылки сообщений (типа “всем маршрутизаторам данной подсети”, “всем узлам-участникам данной группы”, аудио или видеопрограмм) большой группе слушателей или зрителей. Узлы, которые хотят создать такие группы или присоединиться к созданной группе, высылают запрос или подтверждение, соответственно, используя ***протокол управления группами в Интернет (Internet Group Management Protocol, IGMP),*** являющийся составной частью IP и имеющийся в каждом модуле IP (***RFC 1112*** [46]). Состав участников группы формируется динамически: узел может присоединиться или покинуть ее в любой момент. Один и тот же узел может входить в несколько групп, в одну и ту же

группу могут входить узлы различных подсетей. Группы могут быть постоянными или временными. За постоянными группами закрепляются известные IP-адреса в соответствии с ***RFC 3232*** [77], группа может иметь или совсем не иметь участников. Временные групповые IP-адреса существуют только при наличии хотя бы одного участника. Рассылка сообщений модулем IP происходит шлюзами (маршрутизаторами) до тех пор, пока значение “время жизни” (TTL) больше 1, в противном случае дальнейшая рассылка по сети прекращается. IGMP для групповой рассылки использует адреса класса D в диапазоне от 224.0.0.0 до 239.255.255.255. Имеется ряд служебных адресов, например, адрес 224.0.0.1 носит смысл “всем узлам групп своей сети”. Группового адреса для всей глобальной сети Интернет не существует.

Для реализации групповой рассылки дейтаграмм необходимо произвести программное расширение функций в различных модулях. Во-первых, высший уровень должен определить для IP-модуля “время жизни” групповой дейтаграммы, в противном случае оно в модуле IP устанавливается по умолчанию со значением “1” и рассылка происходит только в пределах локальной сети. В адрес назначения подставляется групповой адрес, а в адрес источника – свой индивидуальный IP-адрес. Заказ на присоединение (или отсоединение) от приложения высшего уровня хоста поступает в IP-модуль с указанием IP-адреса группы и номера интерфейса. IP-модуль вносит индивидуальный IP-адрес в список участников группы, после чего поступившая из канала связи дейтаграмма с групповым адресом доставляется вышележащему уровню. Поступившая дейтаграмма уничтожается, если ее групповой адрес не совпадает с номером и содержимым списка, никаких сообщений ICMP не высылается. Во-вторых, должна быть предусмотрена процедура перекодирования группового IP-адреса в групповой MAC-адрес. Для этого в 23 последних бита MAC-адреса 01:00:5E:00:00:00 источника рассылки подставляется 23 последних бита IP-адреса. Например, сетевая служба времени с IP-адресом 224.0.1.1 посылается по MAC-адресу 01:00:5E:00:01:01. Заказ на доставку (или отсев) групповых пакетов поступает в IP-модуль от вышележащего уровня с указанием IP-адреса группы.

IGMP расположен в модуле IP, используя номер протокола 2 (Prot=2). Местоположение и наименование полей IGMP изображено на рис. 2.19. Порядок передачи бит кадра: слева направо/снизу вверх. Назначение полей пакета следующее.

<table>
<tr><td colspan="5">Групповой адрес (Group Address)</td></tr>
<tr><td>Версия
(Version)</td><td>Тип операции
(Type)</td><td>Не используется.
(Unused)</td><td colspan="2">Контрольная сумма сообщения
(Checksum)</td></tr>
<tr><td colspan="5">Адрес назначения (Destination Address)</td></tr>
<tr><td colspan="5">Адрес источника (Source Address)</td></tr>
<tr><td colspan="2">Время жизни
(Time to Live)</td><td>Протокол
(Protocol = 2)</td><td colspan="2">Контрольная сумма заголовка
(Header Checksum)</td></tr>
<tr><td colspan="3">Идентификатор (Identification)</td><td>Флаги
(Flags=0)</td><td>Смещение фрагмента
(Fragment Offset = 0)</td></tr>
<tr><td>Версия
(Version)</td><td>Длина
(IHL)</td><td>Тип услуги
(Type of Service=0)</td><td colspan="2">Общая длина пакета
(Total Length)</td></tr>
<tr><td colspan="5">2 уровень</td></tr>
<tr><td colspan="5">1 уровень</td></tr>
</table>

Рис. 2.19. Местоположение в стеке протоколов и формат IGMP

Версия (Version) – принимает значение 1.

Тип операции (Type) – в это поле помещается код операции (1 – запрос о наличии участников группы, 2 – отчет о наличии участников в группе).

Неиспользуемое поле (Unused) – длина 1 байт, все “0”.

Контрольная сумма сообщения IGMP (Checksum) – используется для обнаружения ошибок при передаче IGMP-пакета, длина 16 бит. Сумма всех 2-байтных слов сообщения полей IGMP совместно с контрольной суммой дает результат 0xFFFF.

Групповой адрес (Group Address) – это поле заполняется групповым адресом при передаче отчета, устанавливается в “0” при передаче запроса.

Свое намерение о вхождении во временную группу хост инициирует рассылкой IGMP-отчета (Type=2) локальному маршрутизатору, который ретранслирует его по предопределенному интерфейсу к серверу услуг по протоколу PIM (Protocol Independent Multicast). Сервер услуг начинает передавать UDP-пакеты с групповым IP-адресом. Каждый маршрутизатор на пути следования IGMP-пакета периодически (не чаще 1 раза в минуту) проверяет существование групп рассылкой по предопределенному интерфейсу IGMP-запроса (Type=1) о наличии участников в группе. Узлы, принадлежащие какой либо группе, в ответ на запрос высылают отчет (Type=2) о своем участии в группе с указанием адреса этой группы. Для избегания одновременной передачи отчетов каждый участник группы высылает IGMP-отчет в случайном диапазоне времени от 0 до 10 секунд после получения запроса. Адрес назначения принимает значение группового адреса. При отсоединении от группы всех участников оконечные устройства перестают отвечать на запросы, на основании чего маршрутизаторы узнают о распаде временной группы и перестают рассылать запросы и ретранслировать поступающие IP-пакеты с групповым адресом.

2.3.4. Протокол Internet версии 6

Изменения IPv4 на IPv6 преследовало следующие цели.

- Расширение адресного пространства. Адресное поле протокола расширено с 32 бит до 128 бит. Необходимость расширения в большой степени вызвано общим увеличением числа пользователей, а также потребностью иметь несколько адресов. Например, мобильные пользователи используют два IP-адреса: один – домашний и другой – временный, по месту текущего месторасположения. Когда устанавливается соединение с мобильным пользователем, то сначала устанавливается соединение на его домашний IP-адрес, откуда оно переадресовывается на текущее местоположение и с него поступает ответ с временным IP-адресом.
- Упрощение формата заголовка.
- Упрощение опциональных полей.
- Возможности разделения потоков.
- Аутентификация и защита данных.

Местоположение и наименование полей ***протокола Интернет версии 6 (Internet Protocol, IPv6, RFC 2460*** [68] изображено на рис. 2.20. Порядок передачи бит кадра: слева направо/снизу вверх. Назначение полей пакета следующее.

Версия (Version) – определяет формат заголовка, длина 4 бита, для версии 4 Version=6 (двоичный код 0110).

Класс трафика (Traffic Class) – используется узлами (оконечными устройствами и маршрутизаторами) для идентификации видов трафика и приоритета их обслуживания, длина 8 бит. Заменяет поле *Тип услуги (Type of Service)* в версии IPv4.

Узлы вправе игнорировать заданный класс обслуживания, если они его не поддерживают.

<table>
<tr><td colspan="4">Расширенный заголовок (Extension headers)</td></tr>
<tr><td colspan="4">Адрес назначения (Destination Address)</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4">Адрес источника (Source Address)</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="2">Длина поля данных (Payload Length)</td><td>Следующий заголовок (Next Header)</td><td>Предел переходов (Hop Limit)</td></tr>
<tr><td>Версия (Version)</td><td>Класс трафика (Traffic Class)</td><td colspan="2">Метка потока (Flow Label)</td></tr>
<tr><td colspan="4">2 уровень</td></tr>
<tr><td colspan="4">1 уровень</td></tr>
</table>

Рис. 2.20. Местоположение в стеке протоколов и формат IPv6

Метка потока (Flow Label) – позволяет исходящему узлу помечать потоки для специального обслуживания, длина 20 бит. Все проходящие узлы должны обслуживать помеченный поток одинаковым образом. Узлы, не поддерживающие эти возможности, должны установить значения всех бит в “0”.

Длина поля данных (Payload Length) – определяет длину октетов, следующих за заголовком, длина 16 бит.

Следующий заголовок (Next Header) – содержит указатель на протокол вышележащего уровня (TCP, UDP, …), либо на наличие опции, расположенной в поле *Расширенный заголовок (Extension headers)*, длина 8 бит.

Предел переходов (Hop Limit) – определяет предельный срок перемещения пакета по сети, длина 8 бит. Устанавливается на передаче, каждый маршрутизатор вычитает из текущего значения единицу, при значении 0 пакет уничтожается.

Адрес источника (Source Address) – определяет сетевой адрес (IP-адрес) отправителя пакета, длина 128 бит.

Адрес назначения (Destination Address) – определяет сетевой адрес получателя, длина 128 бит.

Расширенный заголовок (Extension headers) – содержит опции для использования проходящими узлами и данные вышележащему уровню. Поле *Расширенный заголовок* обрабатывает только тот узел, адрес которого указан в поле *Адрес назначения*. Исключение составляет опция заголовка *От узла к узлу*, расположенная сразу же после основного заголовка, которую должны обрабатывать все проходящие узлы. Наличие этой опции идентифицируется “нулевым значением” поля *Следующий заголовок* (Next Header) основного заголовка. Такая процедура обработки заголовка позволяет существенно снизить время обработки пакета и ускорить доставку пакета до потребителя за минимальное время. Обработка полей ведется в строгом порядке их расположения.

В протоколе IPv6 отсутствует протокол ARP. Его функции возложены на новый протокол обнаружение соседей (Neighbor Discovery Protocol), который переносится протоколом ICMP со следующими сообщениями.

Router solicitation (запрос маршрутизатора, тип сообщения 133). Посылается хостом маршрутизатору после автоконфигурации локального адреса (см. 2.2.6).

Router advertisement (извещение маршрутизатора, тип сообщения 134). Маршрутизатор периодически или по запросу хоста оповещает о префиксе сети, числе переходов, максимальной длине пакета и т.д.

Neighbor solicitation (запрос соседей, тип сообщения 135). Узел посылает это сообщение для запроса канального адреса (MAC-адреса) соседнего узла.

Neighbor advertisement (извещение соседа, тип сообщения 136). Посылается узлом в ответ на запрос соседа с содержанием канального адреса.

Redirect (перенаправление, тип сообщения 137). Посылается маршрутизатором для указания узлу лучшего пути.

Нормальная процедура подсоединения узла к Internet следующая.

- После включения узел производит автоконфигурирование локального адреса, основываясь на MAC-адресе. Например, если MAC-адрес=**34:56:78:9a:bc:de**, то локальный адрес узла fe80:0:0:0:**3656:78**ff:fe**9a:bcde**.
- Узел производит групповую рассылку (*Neighbor solicitation*) с извещением соседей о своем канальном адресе. Если в сети уже имеется еще один узел с таким же канальным адресом, то канальный адрес конфигурируется вручную.
- Если канальный адрес является уникальным, то узел принимает от маршрутизатора (*Router advertisement)* всю информацию и формирует свой глобальный IP-адрес (например, IP_A=2001:1:0:0:**3656:78**ff:fe**9a:bcde**).

Для определения канального адреса B_узла с IP_B=2001:1:0:0:2345:67ff:fe89:abcd A_узел с IP_A=2001:1:0:0:3656:78ff:fe9a:bcde формирует

1. Канальный адрес назначения, используя последние 6 байт IP_B с заменой первых 2 байт на 3333: Destination Link-layer address=33:33:**fe:89:ab:cd.**
2. Групповой адрес назначения (запроса): IP_B=ff02:1:0:0:0:1:ff**89:abcd,** используя префикс ff02:0:0:0:1:ff00:0000 и последние 3 байта IP_B.
3. Канальный адрес источника: MAC_A =34:56:78:9a:bc:de,
4. Индивидуальный адрес источника: IP_A=2001:1:0:0:3656:78ff:fe9a:bcde.
5. Число переходов Hop limit =255, для обнаружения ложных ответных сообщений из другой сети.
6. В поле ICMP: Type=135, MAC_A=34:56:78:9a:bc:de.

В ответ от B_узла получим *Neighbor advertisement* (извещение соседа, тип сообщения 136) с вложенным MAC_B.

2.3.5. Транспортный уровень

Протокол управления передачей (Transmission Control Protocol, TCP) обеспечивает надежный канал обмена данными между прикладными процессами в составной сети, т.е. “из конца в конец”. Протокол ориентирован на установление логических соединений между узлами, обеспечивая следующие возможности (***RFC 793*** [38]).

Надежную доставку сегментов. TCP предохраняет данные от повреждений, утери, дублирования и доставляет пакеты в порядке их формирования на передаче. Это достигается присвоением каждому передаваемому сегменту *порядкового номера (Sequence Number)* и получение *номера подтверждения (Acknowledgement Number)* от принимаемой стороны на каждый правильно принятый сегмент. Когда модуль TCP передает очередной сегмент, то он помещает его копию в очередь и запускает таймер. Если подтверждение на переданный сегмент принято, то сегмент удаляется из очереди. Отсутствие подтверждения означает либо искажение переданного сегмента, либо его утерю, либо потерю подтверждения. Если в течение заданного времени подтверждение не поступает, то не подтвержденный сегмент передается повторно.

Приемник использует порядковые номера для правильной сборки сегментов, которые из сети могут приходить в произвольном порядке. Повреждение передаваемых данных обнаруживается при несовпадении контрольной суммы.

Контроль потока. Это достигается применением механизма *“окна” (Window)*. Принимающий модуль TCP отправляет значение окна передающему модулю. Значение окна задает количество байтов (размер свободного места в приемном буфере), которое принимающий модуль TCP готов принять. В случае правильного приема сегмента, он изымается из буфера и передается вышележащему уровню, размер окна увеличивается на размер изъятого сегмента, передающему модулю передается подтверждение о принятой порции данных и новое значение окна. Нулевой размер окна означает, что приемник не успевает обрабатывать поступающие данные, поэтому отправитель должен приостановить передачу.

Мультиплексирование данных. TCP одновременно может обслуживать несколько прикладных процессов. Каждый прикладной процесс идентифицируется своим уникальным *номером порта*.

Создание независимых соединений. Когда два хоста хотят связаться друг с другом, то они устанавливают соединение. Каждое соединение однозначно специфицируется парой сокетов. При окончании связи соединение разрушается.

Приоритетное обслуживание. TCP опционально может обеспечивать приоритетное обслуживание. По умолчанию оно не используется. Функции приоритетного обслуживания и шифрования данных обычно осуществляется модулем IP или вышележащими уровнями с заранее оговоренными параметрами.

Установление связи между прикладными процессами начинается с того, что один из них вызывает модуль TCP и высылает свои данные в буфер. TCP разбивает поток данных на сегменты, дополняет каждый сегмент управляющей информацией и извещает модуль IP, в какой удаленный модуль TCP следует отправлять эти сегменты. Приемный модуль TCP после обработки управляющей информации помещает сегменты в приемный буфер и извещает соответствующий вышележащий уровень.

Передающий модуль IP упаковывает сегменты в пакеты локальной сети и в зависимости от адреса назначения маршрутизирует их в локальный IP модуль или в исходящий шлюз. Входящий шлюз следующей локальной сети анализирует, следует ли пакет направить конечному узлу или следующему шлюзу. Для передачи по локальной сети входящий модуль IP может разбить принятый пакет на более мелкие части, если локальная сеть использует меньшее значение MTU.

Передающий модуль TCP может накапливать данные и отсылать их, когда это ему удобно, пока от приложения не поступает указание о немедленной отправке (PUSH). В этом случае все неотправленные данные должны быть отправлены немедленно, а при поступлении этого указания приемному модулю TCP, он должен немедленно вывести все принятые наколенные сегменты.

Протокол дейтаграмм пользователя (User Datagram Protocol, UDP) согласно ***RFC 768*** [35] обеспечивает канал обмена данными между прикладными процессами в составной сети, допуская пропадание и дублирование сегментов.

2.3.6. Сеансовый уровень

Рассмотренный ранее протокол маршрутизации OSPF предназначен для маршрутизации IP-пакетов в больших сетях со сложной топологией. Наряду с его очевидными достоинствами (учет петель, учет битов качества обслуживания, высокая устойчивость к изменению топологии сети) он обладает одним недостатком, а именно – высокой вычислительной сложностью. Поэтому чаще всего его применяют в больших автономных сетях и в мощных маршрутизаторах в магистральной сети. Для работы в

небольших автономных сетях с количеством промежуточных маршрутизаторов между узлами не более 15 часто используют более простые протоколы. К таким протоколам относится внутренний ***информационный протокол маршрутизации (Routing Information Protocol, RIP-2)***, учитывающий самую простую метрику – количество проходных маршрутизаторов между узлами. RIP (***RFC 2453*** [67]) представляет собой протокол дистанционно-векторного типа, в качестве критерия целесообразности использования того или иного пути (метрики) используется число переходов (хопов, Hop) через промежуточные маршрутизаторы от узла отправителя до заданного узла назначения. Формализовано задача поиска оптимального пути состоит в просмотре содержимого таблицы маршрутизации и нахождении минимума функции вида

$$D(I, J) = \min_{K} \{d(I, K) + D(K, J)\},$$

где I – номер узла отправителя, J – номер узла назначения, D(I , J) – метрика наилучшего пути между узлами I и J, d(I , K) – метрика между узлом I до соседнего узла K.

Построение таблиц маршрутов происходит следующим образом.

В исходном состоянии администратор сети вручную создает минимальную таблицу маршрутизации, учитывающую только непосредственно подсоединенные сети. После этого маршрутизатор сам начинает посылать сообщения RIP о содержании своей таблицы и получать такие сведения от других маршрутизаторов. Сообщение RIP содержат несколько параметров о каждой известной маршрутизатору сети: IP-адрес, ее маску, метрику. Получив сообщение RIP, маршрутизатор прибавляет к каждой текущей метрике единицу, анализирует новые сообщения с целью помещения их в свою таблицу и рассылает результат другим маршрутизаторам. Сети, имеющую метрику более 15 считаются недостижимыми.

Предположим, что текущий маршрут имеет метрику D и она получена от маршрутизатора G. Если новая метрика, полученная от какого либо другого маршрутизатора лучше, чем D, то текущий маршрут заменяется новым с новой метрикой. Но если новая метрика получена от того же маршрутизатора, то текущий маршрут заменяется всегда, даже если он имеет худшую метрику. Таким образом, в таблице маршрутов хранится информация только о наилучших (с точки зрения принятого критерия) путях.

Рассылку таблиц каждый маршрутизатор осуществляет с периодом 30 секунд, используя не надежный протокол UDP, поэтому сообщения RIP могут теряться. Однако если через 180 секунд сообщение от какого либо маршрутизатора не поступает, то он считается недоступным, а вся информация о маршрутах, полученная от него, считается недействительной и стирается. Чтобы избежать “зацикливания”, маршрутная информация, хранящаяся в таблице, никогда не передается тому маршрутизатору, от которого она получена. При выходе из строя отдельного маршрутизатора из-за конечного времени распространения маршрутной информации могут возникнуть некоторые периоды нестабильной работы, если не принять специальных мер.

RIP-сообщения передаются в дейтаграммах UDP. Каждый узел, использующий RIP, запускает свои процессы передачи и приема маршрутной информации через порт 520. Оконечные устройства, обычно используют RIP пассивно, только принимая сообщения для определения активных шлюзов и для составления своей таблицы маршрутов.

Протокол динамической конфигурации оконечных устройств (Dynamic Host Configuration Protocol, DHCP) решает две задачи: назначение параметров конфигурации и назначение IP-адреса оконечным устройствам (ОУ). Используется модель клиент-сервер, где под понятием клиент подразумевается узел, использующий

DHCP для запроса конфигурации, а под понятием сервер – узел, возвращающий по запросу конфигурационные параметры. Основой DHCP (***RFC 2131*** [58]) является протокол удаленной загрузки узла (Bootstrap Protocol – BOOTP). Сервер и клиент в большинстве сетях не имеют непосредственной связи, поэтому транзитные узлы (BOOTP Relay Agent) должны уметь ретранслировать сообщения между ними. DHCP-сервер конфигурируется администратором сети и ведет централизованное назначение сетевых адресов, что исключает их дублирование. DHCP использует три механизма назначения адреса. При *автоматическом* назначении DHCP-сервер без вмешательства оператора из пула (диапазона) разрешенных адресов возвращает один и тот же адрес без ограничения срока жизни. При *динамическом* назначении – DHCP-сервер возвращает один и тот же адрес в течение ограниченного времени (срока аренды), при *ручном* – DHCP-сервер возвращает один и тот же адрес, назначенный администратором сети. Только механизм динамического назначения адреса позволяет автоматически присваивать тот же IP-адрес другому клиенту, после того, как он оказался не нужен предыдущему клиенту. Это свойство используется для временного подключения клиентов к сети и экономии IP-адресов. При истечении срока аренды клиент обязан пройти процедуру продления конфигурационных параметров.

В качестве транспортного уровня DHCP использует UDP. Клиентский узел, использующий DHCP, запускает свой процесс через порт 68, а сервер – через порт 67.

Для установления соответствия между символическими именами пользователей и их сетевыми адресами в Интернет используется ***система доменных имен (Domain Name System, DNS)***. Доменное имя (***RFC 1034*** [44], ***RFC 1035*** [45]) представляет собой конечную последовательность из N этикеток, разделенных точкой, вида <этикетка N>.<этикетка N-1>.<…>.<этикетка 1.>.< **.** >. Каждая этикетка может состоять из букв, цифр или тире 7-разрядного кода ASCII, причем

нет различия между заглавными и прописными буквами;

этикетка всегда начинается с буквы, а заканчиваться буквой или цифрой;

длина этикетки не должна превышать 63 октетов (символов), а общая длина доменного имени – не более 255 октетов,

последняя этикетка не имеет имени и обозначается “точкой”.

Доменное имя – это перечень этикеток от узла к корню. Домен с этикеткой <N> является частью домена (поддоменом) по отношению к домену с этикеткой <N-1> и носит название зоны. Архитектура системы доменных имен имеет древовидную структуру, аналогичную системе каталогов в компьютере (рис. 2.21).

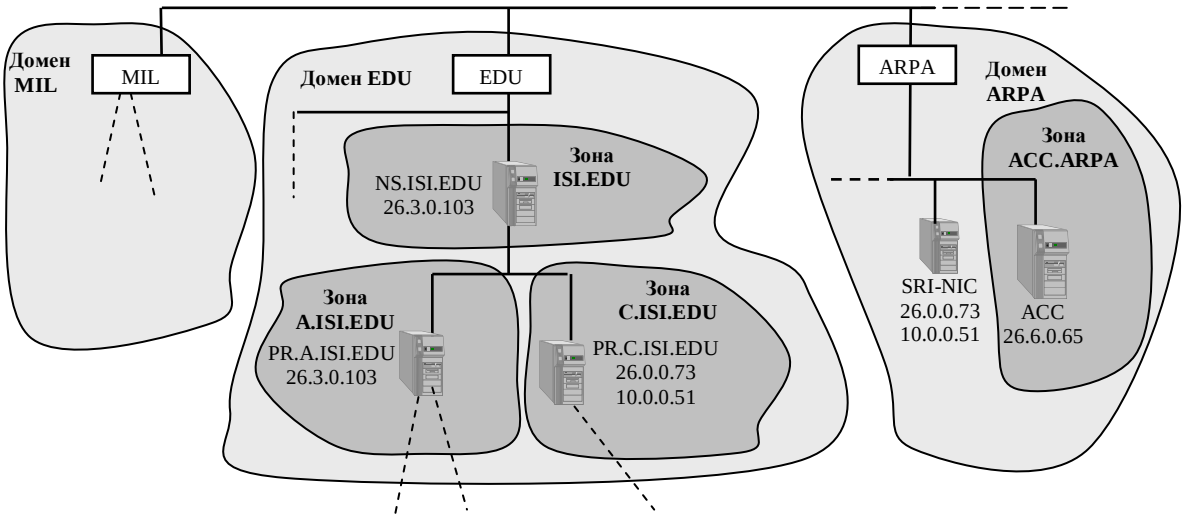

Рис. 2.21. Пример архитектуры системы DNS

Для каждой зоны на сети создается хотя бы один первичный (master) DNS-сервер, хранящий таблицу соответствия “доменное имя – описание ресурса”. Он имеет составленную администратором базу данных зоны, которая представляет собой таблицу соответствия доменных имен и сетевых ресурсов собственной зоны, а также таблицу соответствия символьных имен и сетевых ресурсов некоторых других зон. Данные о собственной зоне являются авторитетными и хранятся на локальном диске в собственном DNS-сервере в виде мастер файла (Master file). Для обеспечения надежности и снижения нагрузки на первичный сервер может дополнительно существовать несколько вторичных DNS-серверов, которые периодически копируют данные из первичного сервера.

Функционирование службы доменных имен поясняет рис. 2.22. Прикладные программы хоста взаимодействуют с DNS-сервером имен через распознаватель имен по модели “клиент-сервер”. Распознаватель имен, получив запрос от прикладной программы, просматривает собственную кэш-память и если не находит в ней требуемые данные, то осуществляет запрос местному DNS-серверу о разрешении доменного имени в IP-адрес. После получения ответа он помещает принятую информацию в кэш-память, а затем, если истекает срок ее действия, то она стирается.

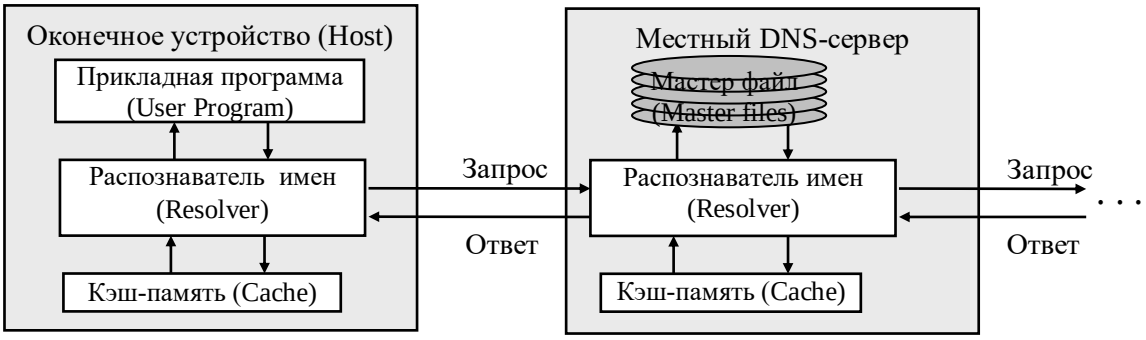

Рис. 2.22. Архитектура службы доменных имен

При получении запроса местный DNS-сервер просматривает либо собственный диск, либо собственную кэш-память и если не находит в них требуемые данные, то осуществляет запрос к следующему DNS-серверу, используя рекурсивную или не рекурсивную процедуру поиска.

Предположим для нашего случая, что серверу зоны ISI.EDU требуются данные о хосте в зоне ACC.ARPA. В случае рекурсивной процедуры сервер зоны ISI.EDU обратится к серверу зоны (домена) EDU (запрос двигается в направлении “снизу вверх”). Сервер зоны EDU найдет в своих записях о ресурсах адрес сервера домена ARPA и перешлет запрос ему. Сервер зоны ARPA найдет в своих записях о ресурсах адрес сервера зоны ACC.ARPA и перешлет запрос ему (теперь запрос двигается в направлении “сверху вниз”). Сервер ACC.ARPA вернет сведения о запрошенных ресурсах, который будет двигаться по проложенной цепочке в обратном направлении. В данном случае местный сервер производит один единственный запрос, а ответ будет авторитетный, т.к. получен от первоисточника ресурсов.

При использовании не рекурсивной процедуры на запрос сервера зоны ISI.EDU сервер зоны EDU вернет адрес следующего сервера, куда следует обратиться, т.е. адрес сервера домена ARPA. Сервер домена ARPA на второй запрос вернет серверу зоны ISI.EDU адрес сервера ACC.ARPA и только на третий запрос от сервера зоны ACC.ARPA будут получены требуемые данные.

Может случиться так, что при обращении к серверу домена ARPA у него сохранилась в кэш-памяти требуемая информация от предыдущего запроса. В этом случае он не станет делать запрос серверу зоны ACC.ARPA, а перешлет эти данные из кэш-памяти. Но в этом случае ответ будет содержать флаг неавторитетного ответа, т.е. запрошенная информация взята из кэш-памяти.

Все элементы системы доменных имен при взаимодействии между собой используют ***протокол разрешения доменных имен***. В качестве транспортного уровня DNS-сервера обычно используют UDP, но для длинных сообщений могут использовать TCP. Клиентский узел и DNS-сервер обмениваются информацией через порт 53.

2.3.7. Представительный уровень

На представительном уровне базовые протоколы Интернет пользователя отсутствуют. Этот уровень, в основном, используется администраторами для обеспечения безопасности сетей.

2.3.8. Прикладной уровень

Протокол Telnet (***RFC 854*** [41]) обеспечивает двухсторонний симметричный обмен байт-ориентированных управляющих сообщений между двумя узлами, каждый из которых представляется как сетевое виртуальное устройство (Network Virtual Terminal, NVT). Один из узлов обычно является сервером, другой – пользователем. Пользователем называется узел, инициировавший соединение, он имеет устройство ввода (клавиатуру) для запроса данных от удаленной стороны и устройство вывода (дисплей или принтер) для отображения запрашиваемых данных.

В качестве транспортного уровня Telnet использует TCP. Пользователь, использующий Telnet, запускает свой процесс через порт 23, набирая на клавиатуре “telnet <IP-адрес/доменное имя>”. На прикладном уровне над Telnet находится либо программа поддержки реального терминала (на стороне пользователя), либо прикладной процесс в обсуживающей машине (на стороне сервера), к которому осуществляется доступ с терминала. Взаимодействие пользователя с сервером происходит по принципу запрос-ответ: пользователь набирает на клавиатуре командную строку и нажимает клавишу “Ввод” (“Enter”), а сервер в ответ высылает соответствующий отклик.

Кодировка запросов и ответов производится в коде ASCII 7-разрядным кодом (первый бит всегда устанавливается в 0). Коды отдельных байтов приведены в табл. 2.11.

Табл. 2.11 – Коды ASCII

	(0) 000	(1) 001	(2) 010	(3) 011	(4) 100	(5) 101	(6)110	(7) 111
(0) 0000	**NUL**	**DLE**	**SP**	**0**	**@**	**P**	**'**	**p**
(1) 0001	**SOH**	**DC1**	**!**	**1**	**A**	**Q**	**a**	**q**
(2) 0010	**STX**	**DC2**	**“**	**2**	**B**	**R**	**b**	**r**
(3) 0011	**ETX**	**DC3**	**#**	**3**	**C**	**S**	**c**	**s**
(4) 0100	**EOT**	**DC4**	**$**	**4**	**D**	**T**	**d**	**t**
(5) 0101	**ENQ**	**NAK**	**%**	**5**	**E**	**U**	**e**	**u**
(6) 0110	**ACK**	**SYN**	**&**	**6**	**F**	**V**	**f**	**v**
(7) 0111	**BEL**	**ETB**	**'**	**7**	**G**	**W**	**g**	**w**
(8) 1000	**BS**	**CAN**	**(**	**8**	**H**	**X**	**h**	**x**
(9) 1001	**HT**	**EM**	**)**	**9**	**I**	**Y**	**i**	**y**
(a) 1010	**LF**	**SUB**	*****	**:**	**J**	**Z**	**j**	**z**
(b) 1011	**VT**	**ESC**	**+**	**;**	**K**	**[**	**k**	**{**
(c) 1100	**FF**	**IS4**	**,**	**<**	**L**	****	**l**	**|**
(d) 1101	**CR**	**IS3**	**-**	**=**	**M**	**]**	**m**	**}**
(e) 1110	**SO**	**IS2**	**.**	**>**	**N**	**^**	**n**	**~**
(f) 1111	**S1**	**IS1**	**/**	**?**	**O**	**_**	**o**	**DEL**

Для определения двоичного кода символа следует к коду колонки приписать код строки, а для определения шестнадцатеричного – к значению кода колонки приписать значение кода строки. Например, прописной букве “а” соответствует двоичный код 0110 0001 или шестнадцатеричный – 61, цифре “7” – двоичный код 0011 0111 или шестнадцатеричный – 37 и т. д. Коды 0 – 31 являются служебными и используются для операциями над текстом: NUL(0) – незначимая комбинация; SOH(1) – начало заголовка; STX(2) – начало текста (после заголовка); ETX(3) – конец текста; EOT(4) – конец передачи текста; ENQ(5) – запрос идентификатора или статуса удаленного терминала; ACK(6) – подтверждение приема; BEL(7) – указание о включении аварийного (извещающего) сигнала; BS(8) – возврат каретки (курсора) на предыдущую позицию; HT(9) – горизонтальная табуляция; LF(10) – перевод курсора на следующую стоку той же позиции; VT(11) – вертикальная табуляция; FF(12) – перевод курсора на следующую страницу в той же позиции; CR(13) – перевод курсора в начало строки; …

Протокол Telnet позволяет пользователю только управлять операциями на удаленном узле, в том числе читать названия и удалять из него файлы, но не обеспечивает транспортировку файлов между разными узлами. Эту функцию обеспечивает ***протокол пересылки файлов (File Transfer Protocol, FTP)***. В качестве транспортного уровня FTP (***RFC 959*** [43]) использует TCP. Для транспортировки файлов вначале создается канал управления по протоколу Telnet (по умолчанию порт 21), который используется для передачи команд от пользователя и получения откликов от сервера. Затем создается канал передачи данных (по умолчанию порт 20), который открывается и закрывается по мере необходимости в течение сеанса связи. Канал управления закрывается только после закрытия информационного канала.

Простейший протокол пересылки файлов (Trivial File Transfer Protocol, TFTP) используется для дистанционной загрузки программного обеспечения и конфигурационных файлов, например, в шлюзы и маршрутизаторы по их запросу, который они инициируют при включении питания. В качестве транспортного уровня TFTP использует UDP с портом 69. TFTP (***RFC 1350*** [51]) не требует идентификации пользователя (запроса имени пользователя и пароля). Загрузка производится блоками по 512 байт. Надежность доставки обеспечивается TFTP.

Простой протокол передачи почты (Simple Mail Transfer Protocol, SMTP) согласно ***RFC 821*** [39] в качестве транспортного протокола использует TCP (порт 25) и применяется для пересылки сообщений от клиентов к почтовым серверам клиентов и пересылки сообщений между серверами. Адресация почтовых сообщений аналогична той, что принята в система доменных имен (Domain Name System, DNS). Почтовый адрес имеет вид Alex@spez.com, где Alex – символическое имя, spez.com – почтовый домен.

Вначале по инициативе пользователя устанавливается TCP-соединение от клиента Alex к своему почтовому серверу (mail2.spez.com) и поверх него по протоколу SMTP пересылается почтовое сообщение. Признаком окончания обмена является передача “точки”. После передачи сообщения TCP-соединение разрушается. Для получения IP-адреса с именем получателя Bob почтовый сервер mail2.spez.com выполнит запрос (иногда несколько запросов) к DNS-серверу и получит IP-адрес почтового сервера mail3.param.ru, к которому приписан абонент Bob. Таким же образом почтовое сообщение будет доставлено почтовому серверу mail3.param.ru.

Протокол SMTP обеспечивает только доставку почтовых сообщений до почтового сервера. Забрать же сообщение из своего почтового ящика абоненту Bob позволяет ***протокол обслуживания почтового офиса (Post Office Protocol, POP)*** версии 3 (POP3). Протокол POP3 согласно ***RFC 1939*** [56] функционирует поверх протокола TCP по порту 110.

Протокол передачи гипертекста (Hyper Text Transfer Protocol, HTTP), описанный в RFC 2616 [71], является настолько популярным протоколом, что у многих он ассоциируется с самой сетью Интернет. Большинство приложений используют модель “клиент-сервер”, основанной на Web-технологии. В соответствии с ней на сервере размещаются Web-документы, которые интерпретируются программой навигации, функционирующей на рабочей станции клиента.

Логически Web-документ представляет собой документ, имеющий ссылки на различные Web-страницы, каждая из которых может в свою очередь содержать ссылки на другие объекты и Web-документы, расположенные в любом узле сети. Web-документ, являясь по виду аналогом обычного “бумажного” документа, имеет ссылки на другие объекты, выделенные другим цветом и подчеркиванием. Выделенные объекты имеют невидимую на экране пользователя привязку к универсальному локатору ресурса (Uniform Resource Locator, URL), представляющий собой определенный формат записи вида ***схема://имя_узла/описание***. Наиболее часто применяемые *схемы* в URL следующие – http (ссылка на Web-страницу), mailto (ссылка на почтовый адрес), ftp (ссылка на FTP-сервер), file (файл на компьютере). *Имя узла* определяет местоположение компьютера, где расположен ресурс, например, /intel.com/. Наконец, третьей составляющей является *описание*, где отображается путь к ресурсу на данном компьютере, например, *products/desktop/index.htm* (в данном случае на файл с именем *index.htm,* расположенный иерархически в директориях *products* и *desktop*). Ссылки могут быть не только на текст, но и на изображения, видеофильмы, музыкальные файлы и т.д. Наиболее часто применяемые расширения файлов в URL следующие – html, htm (Web-страница), gif, jpg (изображение), wav (музыкальный файл), txt (текстовый файл), zip (сжатый файл), exe (исполняемая программа).

Для написания Web-документа используется специальный гипертекстовый язык меток – Hyper Text Markup Language, HTML – текст, состоящий из ASCII-символов. Этот язык разметки состоит из команд форматирования, позволяющий установить положение объекта на странице, изменить размер, цвет, расстояние между строками, сделать метку и т.д. Метка, означающая наличие связи, имеет вид <a> и происходит от слова anchor (якорь). В общем виде ссылка имеет следующий формат: * текст *, где *url* (universal resource locator) в простейшем случае может быть именем файла. Текст обозначает действительный текст в документе, который может быть подсвечен, выделен другим цветом или помечен цифрой. Этот текст говорит программе просмотра, что в URL можно найти связанную с данным документом информацию или изображение. Для вызова этой информации или изображения на экран достаточно указать мышкой на текст и нажать кнопку.

Если URL указывает на объект, находящийся в другой сети, эта процедура может занять некоторое время. Метка может использоваться и для ссылки на определенный раздел документа: * текст *, где *refname* является меткой в вашем документе. В общем случае URL указывает тип и место расположения ресурса: *сервер://host.domain[:port]/path/объект*, где в качестве сервера может стоять: FTP, Telnet, HTTP, Gopher, Wais, News. Path описывает проход к каталогу, где лежит объект.

Расширяемый язык разметки (Extensible Markup Language, XML) рассматривается как следующее поколение HTML. Наряду с командами форматирования, похожие на HTML, в нем предусмотрены возможности определения новых команд, которые определяет сам пользователь.

Архитектура управления IP-сети представляет собой совокупность управляющих станций и сетевых элементов. Под сетевыми элементами понимаются хосты (терминалы), шлюзы, серверы и тому подобные устройства, имеющие приложения агентов управления, функционирующие под управлением управляющей станции.

Простой протокол управления сетью (Simple Network Management Protocol, SNMP) приведен в ***RFC 1155*** [47] и предназначен для переноса управляющей информации между сетевыми элементами и управляющей станцией (компьютером администратора).

Основными задачами протокола являются доставка запрашиваемой информации от сетевых элементов к управляющей станции и установка в элементах конфигурационных параметров, заданных управляющей станцией. В качестве транспортного протокола используется протокол UDP. Узел принимает сообщения на порт 161, а отсылает на динамический порт управляющей станции. Сообщения об аварийных ситуациях узел передает на порт 162 управляющей станции.

Полная информация о сетевом элементе (***RFC 1213*** [48]) храниться в его индивидуальной базе данных (Management Information Base, MIB), которая описывается в форме языка абстрактно-синтаксических нотаций версии 1 (Abstract Syntax Notation One, ASN.1) согласно ***RFC 4750*** [84]. Каждый параметр сетевого элемента (объекта) в MIB имеет свой идентификатор (OBJECT IDENTIFIER), синтаксис и метод кодировки. Содержание MIB администрируют различные международные организации.

Идентификация параметров объекта в MIB происходит через дерево вложенных каталогов. Каждый каталог имеет свое имя (label), представляющее собой короткий текст или соответствующее ему целое число. Корневой каталог не имеет имени и имеет три подкаталога: первый – ccitt (0) – администрирует международный комитет по телеграфии и телефонии (МККТТ), второй – iso (1) – администрирует международная организация по стандартизации, третий – joint-iso-ccitt (2) – находится в общем ведении обеих организаций. Под каталогом iso (1) имеется каталог org (3) для размещения описаний объектов других международных организаций, в том числе, министерства обороны США (dod), имеющего подкаталог Internet (1). Таким образом, все объекты Интернет имеют общий префикс 1.3.6.1.

ВОПРОСЫ К РАЗДЕЛУ 2.3

1. Какую тактовую частоту используют узлы при асинхронном обмене?

Ответ. Каждый из них используют свою тактовую частоту.

2. Какую тактовую частоту используют узлы при синхронном обмене?

Ответ. Приемник использует тактовую частоту передатчика.

3. Почему между узлами не используется код NRZ?

Ответ. Из-за трудности выделения приемником тактовой частоты.

4. Какую адресную информацию используют повторители и концентраторы для пересылки пакетов с одного сегмента сети на другой?

Ответ. Никакую.

5. Какую адресную информацию используют мосты для пересылки пакетов с одного сегмента сети на другой?

Ответ. MAC-адрес.

6. Какую адресную информацию используют коммутаторы для пересылки пакетов с одного порта сети на другой?

Ответ. MAC-адрес.

7. Какую адресную информацию используют маршрутизаторы для пересылки пакетов с одного интерфейса сети на другой?

Ответ: IP-адрес.

8. Какие протоколы сетевого уровня используют протокол IP?

Ответ. ICMP с номером протокола 0x0001, IGMP с номером протокола 0x0002, OSPF с номером протокола 0x0059.

9. Какие протоколы используются на транспортном уровне?

Ответ. UDP с номером протокола 0x0011, TCP с номером протокола 0x0006.

10. В каком случае модуль IP осуществляет фрагментацию?

Ответ. В случае если поступивший пакет имеет большую длину (MTU), чем она разрешена на адресуемом интерфейсе.

11. Какую информацию рассылает граничный маршрутизатор зоны во внешнюю сеть?

Ответ. Только состав сетей и метрики их достижения.

12. Каким образом OSPF-маршрутизатор узнает об установлении связи с соседним маршрутизатором?

Ответ. Обнаружением своего IP-адреса в списке сообщения Hello, полученного от соседнего маршрутизатора.

13. Обеспечивается ли надежная доставка сообщений ICMP?

Ответ. Нет.

14. Что делает узел, входящий в групповую рассылку, при приеме IGMP отчета от другого участника группы?

Ответ. Перезапускает таймер отсылки отчета.

15. Каковы основные преимущества протокола IPv6 перед IPv4?

Ответ. Расширение адресного пространства и упрощение процедуры обработки полей протокола.

16. Какие основные функции протокола TCP?

Ответ. Обеспечивает гарантированную доставку, сегментирует длинные данные приложений, обеспечивает сквозное взаимодействие “из конца в конец”.

17. Что означают значения порядковый номер и номер подтверждения в TCP?

Ответ. Условный номер первого байта в передаваемом сегменте, условный номер первого байта в ожидаемом следующем сегменте.

18. Какие основные функции протокола UDP?

Ответ. Обеспечивает не гарантированную доставку данных.

19. Какой транспортный протокол и номер порта использует RIP?

Ответ. Протокол не надежной доставки UDP, порт 520.

20. Чем достигается надежность рассылки RIP-сообщений?

Ответ. Периодической (через 30 секунд) рассылкой.

21. Какое DHCP-сообщение является первым и почему?

Ответ. Первое сообщение DHCPDISCOVER требуется для обнаружения DHCP-сервера.

22. Какие значения принимают MAC-адрес назначения, IP-адрес источника и тип протокола в сообщении DHCPDISCOVER?

Ответ. MAC-адрес=ff:ff:ff:ff:ff:ff, IP-адрес=0.0.0.0, тип протокола 0x0800.

23. Какую основную информацию хранит DNS-сервер?

Ответ. Таблицу соответствия доменных (символьных) имен IP-адресу пункта назначения.

24. Может ли администратор домена def.abc создать учетную запись hh.def.abc и hh.abc?

Ответ. Первую – да, вторую – нет, поскольку администратор домена def.abc не ответственен за домен abc.

25. Назовите основное назначение протоколов FTP и TELNET.

Ответ. FTP предназначен для передачи файлов между узлами, а TELNET – для управления операциями на удаленном узле.

26. Какие и сколько соединений использует FTP?

Ответ. FTP постоянно использует одно TCP-соединение для управления, а на время транспортировки файла временно создается второе TCP-соединение.

27. Перечислите основные отличия протокола FTP от TFTP.

Ответ. а)Для доставки информации протокол FTP использует транспортный протокол TCP, а протокол TFTP – UDP. б)FTP использует для передачи блоки информации переменной длины, а TFTP использует блоки фиксированной длины 512 байт. с)Надежная доставка информации протокола FTP обеспечивается транспортным протоколом TCP, а надежная доставка информации протокола TFTP обеспечивается протоколом приложения.

28. Какие транспортные протоколы используют SMTP и POP3 и почему?

Ответ. Для надежной доставки и фрагментации длинных сообщений используется протокол TCP.

29. На каких участках сети используются протоколы SMTP и POP3?

Ответ. На участке исходящий пользователь – входящий почтовый сервер используется протокол SMTP. На участке входящий почтовый сервер – пользователь используется протокол POP3.

30. Объясните назначение многоцелевого расширения электронной почты в Интернет (MIME).

Ответ. Применяется для правильного воспроизведения каждой части сообщения, каждое из которых может иметь различную кодировку.

31. Назовите состав полей универсального локатора ресурса (URL).

Ответ. Схема (http://, mailto:, ftp://, …), имя сервера (/intel.com/), путь к ресурсу на данном сервере (/products/desktop/index.htm).

32. Назовите язык описания базы данных сетевого элемента.

Ответ. Для описания базы данных сетевого элемента используется язык абстрактно-синтаксических нотаций версии 1 (Abstract Syntax Notation One, ASN.1).

33. Напишите общий префикс каталога объекта Интернет в протоколе SNMP.

Ответ. Все объекты Интернет имеют общий префикс 1.3.6.1.

2.4. Обслуживание мультимедийных приложений

2.4.1. Принципы формирования пакетов

Изначально сеть Интернет проектировалась только как сеть передачи данных и не была рассчитана на передачу приложений в реальном масштабе времени, каковыми являются приложения видео и аудио. Поэтому возникла необходимость дополнить базовые протоколы Интернет дополнительными специализированными протоколами.

Во многих случаях исходная информация является аналоговой. Преобразование аналогового сигнала в цифровой сигнал может осуществляться различными способами. Преобразование обычно включает в себя четыре этапа:

1. *Ограничение полосы (Band Limitation).* Выбирается эффективный спектр речевого сигнала, находящийся в диапазоне от 300 до 3400 Гц.

2. *Квантование (Sampling).* Для правильного воспроизведения аналогового сигнала достаточно передавать его отдельные значения, между которыми исходный аналоговый сигнал изменяется монотонно. Поскольку наибольшие изменения создает самая высокая частота исходного аналогового сигнала, то частота квантования (выборки) выбирается равной удвоенному значению наивысшей частоты исходного аналогового сигнала, т.е. не менее 6800 Гц, в практических реализациях она принята равной 8000 раз в секунду.

3. *Кодирование (Coding).* Каждое значение выборки измеряется и кодируется в двоичном коде длиной 8 бит (1 байт). Первый бит определяет знак выборки (0 – отрицательное значение, 1 – положительное). Следующие три бита определяют сегмент, т.е. часть масштабной области, в которой находится измеренная выборка (000 –

младший сегмент, 111 – старший). Последние четыре бита определяют один из 16-и шагов внутри сегмента (0000 – первый шаг, 1111 – последний). Погрешность измерения порождает шумы квантования. Таким образом, для речевого сигнала полоса передачи равна 8000 $с^{-1}$ · 8 бит = 64000 бит/с или 64 Кбит/с.

4. *Сжатие цифрового сигнала.* Эта операция производится над группой оцифрованных выборок и не является обязательной, если в сети используются высокоскоростные каналы связи. Однако некоторые части сети могут иметь гораздо более низкую скорость передачи. В этих случаях сжатие цифрового сигнала становится необходимым. Следует отметить, что сжатие сигнала на передаче всегда связано с понижением качества его воспроизведения на приеме. Для оценки передачи качества голосового сигнала ***Рекомендация ITU-T P.800*** [22] использует шкалу средней экспертной оценки (Mean Opinion Scores, MOS), приведенную в табл. 2.12.

Табл. 2.12 – Шкала MOS

MOS-оценка	Описание качества
5	Междугородное качество, соответствующее качеству передачи по медным физическим проводам
4	Качество, близкое к междугородному качеству с почти незаметными искажениями
3	Ощутимые искажения, приводящие к повторению части сказанного
2	Искажения весьма раздражают
1	Невозможно вести разговор

Группа оцифрованных выборок (со сжатием или без) объединяются в поле данных, обрамляются заголовками стека протоколов и в виде пакета передаются в канал связи. На приеме восстановление аналогового сигнала производится в обратном порядке.

В соответствии с ***Рекомендацией G.711*** [9] время между соседними выборками равно 125 микросекунд, поэтому в поле данных пакета (Payload) за 10, 20 или 30 мс помещается соответственно 80, 160 или 240 байт. При прохождении по стеку протоколов добавляются заголовки RTP (12 байт), UDP (8 байт), IP (20 байт) и байты канального уровня (8 байт для PPP или 14 байт для Ethernet). В наихудшем случае (Ethernet) с учетом преамбулы и межпакетного интервала в сумме добавляется 78 байт (рис. 2.23).

Preamble (8 байт)	Ethernet (14 байт)	RTP, UDP, IP (40 байт)	Payload	CRC (4 байта)	Pause (12 байт)

Рис. 2.23. Поля речевого пакета (Ethernet)

Общая характеристика пакета для приведенных случаев дана в табл. 2.13.

Табл. 2.13 – Параметры пакетов с кодеком G.711

Тип кодека	Алгоритм кодирования	Полоса передачи речи, Кбит/с	Время формирования пакета, мс	Общая длина пакета, байт (*LEN*)	Суммарная полоса передачи, Кбит/с (*bw*)	MOS
G.711	PCM	64	10 (80 байт) 20 (160 байт) 30 (240 байт)	80+78=158 160+78=238 240+78=318	126.4 95.2 84.8	4.3 4.3 4.3

Рекомендация G.729A [11] использует алгебраический линейный предиктивный кодер связанной структуры (Conjugate-Structure Algebraic-Code-Excited Linear-Prediction, CS-ACELP). Для сжатия речевых пакетов применяются специализированные цифровые процессоры (Digital Signal Processor, DSP) высокой производительности, использующие различные алгоритмы сжатия. В качестве исходных данных используется блоки 10 (или

20) миллисекундных выборок размером 80 (или 160) байт, а на выходе процессор формирует пакет длиной 10 (или 20) байт. Время обработки блока процессором – 5 мс, общее время формирования пакета 15 (или 25) мс. Общая характеристика пакета дана в табл. 2.14.

Табл. 2.14– Параметры пакетов с кодеком G.729A

Тип кодека	Алгоритм кодирования	Полоса передачи речи, Кбит/с	Время формирования пакета, мс	Общая длина пакета, байт (*LEN*)	Суммарная полоса передачи, Кбит/с (*bw*)	MOS
G.729A	CS-ACELP	8	15 (10 байт)	10+78=88	70.4	3.9
G.729A	CS-ACELP	8	25 (20 байт)	20+78=98	39.2	3.9

Рекомендация G.723.1A [10] использует два алгоритма: многоимпульсное квантование с максимальным правдоподобием (Multipulse Maximum Likelihood Quantization, MP-MLQ) или алгебраическое линейное предиктивное кодирование (Algebraic-Code-Excited Linear-Prediction, CS-ACELP). В качестве исходных данных используется блоки 30-и миллисекундных выборок размером 240 байт. В первом случае на выходе процессор формирует пакет длиной 192 бит (24 байта), во втором – 160 бит (20 байт). Время обработки блока процессором – 7.5 мс, общее время формирования пакета 37.5 мс. Общая характеристика пакета для приведенных случаев дана в табл. 2.15.

Табл. 2.15 – Параметры пакетов с кодеком G.723.1A

Тип кодека	Алгоритм кодирования	Полоса передачи речи, Кбит/с	Время формирования пакета, мс	Общая длина пакета, байт (*LEN*)	Суммарная полоса передачи, Кбит/с (*bw*)	MOS
G.723.1A	MP-MLQ	6.4	37.5 (24 байта)	24+78=102	27.2	3.8
G.723.1A	ACELP	5.3	37.5 (20 байт)	20+78=98	26.1	3.6

Для еще большей экономии полосы передачи кодеки G.711, G.729, G.723 используют процедуру подавления пауз в разговоре. На передаче детектор активности (Voice Activity Detection, VAD) обнаруживает паузы в речи путем сравнения энергии в пакете, поступившим от источника, с энергией в пакете, поступившим от генератора шума. При обнаружении длительной паузы (длительностью примерно 10 мс) передатчик начинает посылать пакеты (Silence Insertion Descriptor, SID), имеющие гораздо меньшую длину (4 байта), чем обычные речевые пакеты. Приемник, получив такой пакет, включает местный генератор комфортного шума (Comfort Noise Generator, CNG), чтобы снять у абонента ощущение обрыва канала связи. Процедура подавления пауз в разговоре позволяет уменьшить нагрузку на 35-45%.

Рекомендация G.726 [12] предлагает использовать следующие кодеки (табл. 2.16).

Табл.2.16 – Параметры пакетов с кодеком G.726

Тип кодека	Алгоритм кодирования	Полоса передачи речи, Кбит/с	Время формирования пакета, мс	Общая длина пакета, байт (*LEN*)	Суммарная полоса передачи, Кбит/с (*bw*)	MOS
G.726	ADPCM	40	20 (100 байт)	100+78=178	71.2	4.1
G.726	ADPCM	32	20 (80 байт)	80+78=158	63.2	4.1
G.726	ADPCM	24	20 (60 байт)	60+78=138	55.2	4.0
G.726	ADPCM	16	20 (40 байт)	40+78=118	47.2	4.0

Последние два случая (с полосой 24 и 16 Кбит/с) использовать не рекомендуется.

Формирование пакетов видео приведено в Рекомендациях H.261, H.262 и H.263. Рекомендации предназначены для передачи подвижных изображений телевизионного стандарта с разверткой 525 или 625 строк со скоростью развертки от 10 до 75 кадров в

секунду. Длина кадра различна, полоса передачи зависит от размера изображения и от скорости развертки и расположена от 0.064 до 100 Мбит/с. Рекомендации описывают несколько опций сжатия, позволяющие понизить суммарную полосу передачи.

2.4.2. Параметры качества обслуживания

При передаче речи по IP-сети возникают следующие проблемы [1].

- Постоянные задержки прохождения сигналов от передатчика до приемника.

Задержки в любых сетях связи существуют всегда. Абоненты не ощущают задержек и общаются в реальном масштабе времени, если суммарная задержка “из конца в конец” не превышает 150 мс. Оценка влияния задержек на качество обслуживания приведено в ***Рекомендации ITU-T G.114*** [8] (табл. 2.17).

Табл. 2.17 – Влияние сетевой задержки на качество обслуживания

Значение задержки “из конца в конец”	Оценка качества
0 – 150 миллисекунд	Очень хорошее
150 – 300 миллисекунд	Хорошее
300 – 400 миллисекунд	Удовлетворительное
Свыше 400 миллисекунд	Не удовлетворительное

Задержки, во-первых, возникают при формировании пакета. В цифровых сетях интегрального обслуживания с коммутацией каналов (ЦСИО) каждый фрагмент речи длительностью 125 мкс преобразуется в 8 бит (байт) и сразу же передается в заранее установленный канал связи. В сети с коммутацией пакетов (СКП) несколько байтов собираются в один пакет, иногда подвергаются анализу для устранения избыточности (сжимаются до меньшего значения), обрамляются заголовком и хвостовиком и только потом передаются. В результате такой сложной процедуры время формирования пакета *tcod* принимает значение от 10 до 37,5 мс. Некоторое время *tdec* (до 7.5 мс) требуется на приеме для обратных операций.

Во-вторых, имеются задержки переприема (serialization), связанные с тем, что для определения маршрута следования поступившего пакета маршрутизатору необходимо прочитать адресную часть пакета, а для этого необходимо принять весь пакет, или, по крайней мере, его заголовок. Следует отметить, что при наличии очереди задержка переприема определяется полосой передачи *bw*, занимающей часть скорости передачи по каналу связи *VEL*. Поэтому маршрутизатору для приема пакета длиной *LEN* байт от приложения, которому выделена полоса передачи *bw* Кбит/с, может потребоваться время $8 \cdot LEN/bw$ мс. Например, для передачи пакета речи с использованием Рекомендации G.711 (*LEN*=238 байт, *bw*=95.2 Кбит/с) время переприема на одном маршрутизаторе может составить 20 миллисекунд. Поэтому для снижения задержки переприема коммутация в современных маршрутизаторах осуществляется сразу же после приема заголовка пакета.

В-третьих, при перегрузке на сети маршрутизаторы устанавливают поступающие пакеты в выходную очередь. При большой интенсивности трафика создается дополнительная задержка *tr* из-за прохождения очереди на передачу.

В-четвертых, задержки возникают из-за конечного времени распространения сигнала (примерно 6 микросекунд на километр), которое имеет небольшое значение на коротких линиях, за исключением спутниковых каналов связи, где задержка может достигать до 400 мс.

- Потеря пакетов.

В процессе прохождения пакета по сети связи он может быть

- искажен из-за влияния помех;
- доставлен не по адресу из-за сбоя в программном обеспечении или в канале;
- утерян по вышеназванным причинам.

Потеря пакета приводит к различным последствиям в зависимости от способа преобразования аналогового сигнала, причем наиболее катастрофические последствия наступают для пакетов, подвергшихся сжатию. ***Рекомендация ITU-T G.114*** [8] нормирует долю потерянных пакетов на уровне не более 5%.

- Вариация задержки (джиттер).

Дополнительные задержки возникают при прохождении систем коммутации. В системе с коммутацией каналов (ЦСИО) задержка возникает в каждом узле коммутации из-за перестановки входного временного интервала на более позднее значение на выходе узла и имеет постоянное значение (не более 125 мкс на ступень искания) на все время установления соединения. Суммарная постоянная задержка (“из конца в конец”) даже при большом количестве транзитных узлов легко укладывается в требуемую норму tsum=150 мс.

В системе коммутации пакетов (СКП) коммутацию выполняют маршрутизаторы, которые коммутируют поступающие пакеты в порядке их поступления и с учетом приоритета. Случайный процесс поступления пакетов от разных источников приводит к тому, что даже регулярный порядок поступающих на вход пакетов от одного источника, на выходе маршрутизатора нарушается. Это приводит к тому, что в отличие от ЦСИО в СКП возникают дополнительные задержки переменной величины. Для компенсации переменной сетевой задержки приемники пакетов хоста всегда оснащаются сглаживающим буфером, аналогично эластичной памяти в узлах ЦСИО. Поступающие пакеты записываются в момент их приема в очередную ячейку памяти буфера, а считывание пакетов происходит с некоторой постоянной задержкой *tbuf* (со смещением номеров ячеек), определяемой емкостью буфера *Buf*.

Сеть с коммутацией пакетов Интернет предназначена для обслуживания различных типов услуг (видео, речь, данные). Для реализации каждой услуги необходимо выделить определенный сетевой ресурс – производительность управляющих устройств, полосу передачи канала связи и т.д. Для понимания возникновения дефицита канальных ресурсов рассмотрим типовую схему обслуживания услуг маршрутизатором (рис. 2.24).

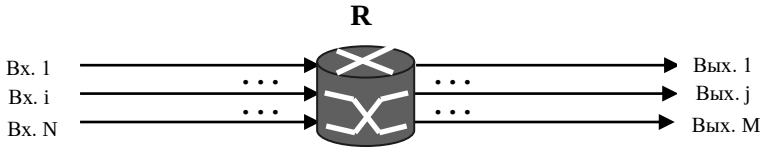

Рис. 2.24. Схема обслуживания вызовов

Каждая услуга характеризуется своей полосой передачи *bw* (Band Width) и длиной пакета *LEN*. Услуга порождает поток пакетов (вызовов). На каждый из N входов (интерфейсов) сетевого маршрутизатора поступает множество пакетов, принадлежащим разным услугам. Каждый пакет может потребовать соединения с одним или несколькими из M выходов (интерфейсов) маршрутизатора. Если в момент поступления пакета имеется свободный требуемый выход, то пакет обслуживается немедленно, в противном случае он обслуживается в порядке очереди. Дефицит канальных ресурсов возникает в моменты времени, когда несколько практически одновременно поступивших пакетов требуют соединения с одним и тем же выходом маршрутизатора, имеющего конечную скорость передачи *VEL*.

Традиционно в Интернет широко использовался способ распределения ресурсов “с наибольшими усилиями” (Best Effort), заключающийся в принципе “первый пришел – первый обслужен” (First In – First Out, FIFO). Такой алгоритм обслуживания хорошо работает, если приложения терпимы к задержкам пакетов из-за образования очередей или в случае, если сетевые ресурсы намного превышают потребности поступающих

вызовов. Например, если имеются высокопроизводительное управляющее устройство и сверхскоростные каналы связи, то аппаратная задержка и время занятия канала связи для передачи пакета ничтожно и очереди в маршрутизаторе не возникают. Такая ситуация показана на рис. 2.25а, когда на один выход (Вых) поступают пакеты от трех входов (Вх.1, Вх.2, Вх.3). Однако, запросы реально поступающих вызовов сопоставимы с возможностями канальных ресурсов, а иногда значительно их превышают. На рис. 2.25б показана та же ситуация при использовании каналов связи с малой скоростью.

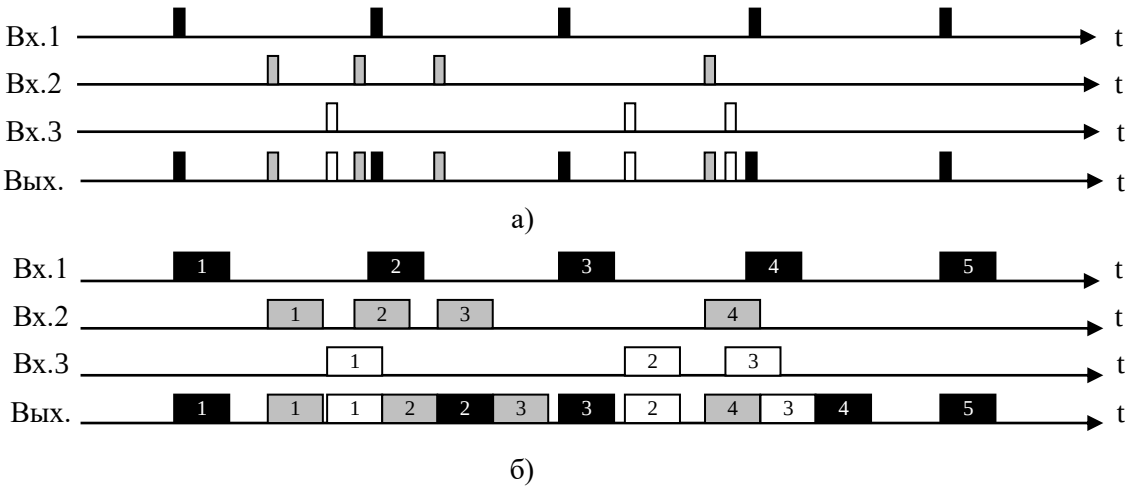

Рис. 2.25. Диаграмма обслуживания пакетов алгоритмом FIFO на а) высокоскоростных и б) низкоскоростных каналах

Видно, что использование низкоскоростных каналов приводит к значительному джиттеру всех потоков, что совершенно неприемлемо для таких приложений, как передача речи. Следовательно, проблема состоит в том, чтобы создать такой порядок обслуживания потоков, чтобы приложения, чувствительные к задержкам, имели большую пропускную способность за счет снижения пропускной способности других приложений, терпимых к задержкам. Это достигается введением специальных механизмов: механизма дозирования и выравнивания трафика; механизма предотвращения перегрузки сети; механизма обслуживания очередей на основе приоритетов.

2.4.3. Механизмы обеспечения качества обслуживания

Для устойчивой работы сети в каждом маршрутизаторе используются механизмы ***дозирования и выравнивания трафика*** (Traffic Shaping). Под дозированием понимается ограничение интенсивности входящего и/или исходящего потока пакетов из узла, а под выравниванием – их равномерное распределение во времени. Эти же механизмы используется в каждом пограничном маршрутизаторе для проверки соответствия интенсивности поступающих пакетов заданному профилю (согласованному значению). Дозатор и выравниватель трафика используют одну и ту же схему – “корзина маркеров” (token bucket). Схема “корзина маркеров” функционирует следующим образом (рис. 2.26).

Каждому интерфейсу ставится в соответствие три параметра: согласованная (средняя) скорость передачи (Committed Information Rate – *r*, байт/с), размер буфера корзины маркеров *b* (байт), размер буфера очереди *B* (байт). Их конфигурирование в маршрутизаторе задается командой ***rate limit input/output*** *r b B*. Генератор маркеров периодически с интервалом времени *T0=b/r* пополняет корзину *b* маркерами (байтами).

На вход дозатора/выравнивателя в буфер очереди емкостью *B* байт поступают пакеты со скоростью *Bin* байт в секунду. Если число маркеров в корзине равно или превышает длину первого в очереди пакета, то такой пакет передается, и соответствующее число маркеров изымается из корзины. При недостатке маркеров в корзине пакет ожидает в буфере очереди, пока в корзине не накопится достаточное количество маркеров. При заполнении корзины приток новых маркеров от генератора прекращается (на рис. 2.26 – черный маркер), при наличии в буфере очереди *B* байт вновь поступающие пакеты отбрасываются. Механизм функционирует нормально, если скорость передачи интерфейса превышает согласованную скорость передачи, т.е. при *VEL>r*.

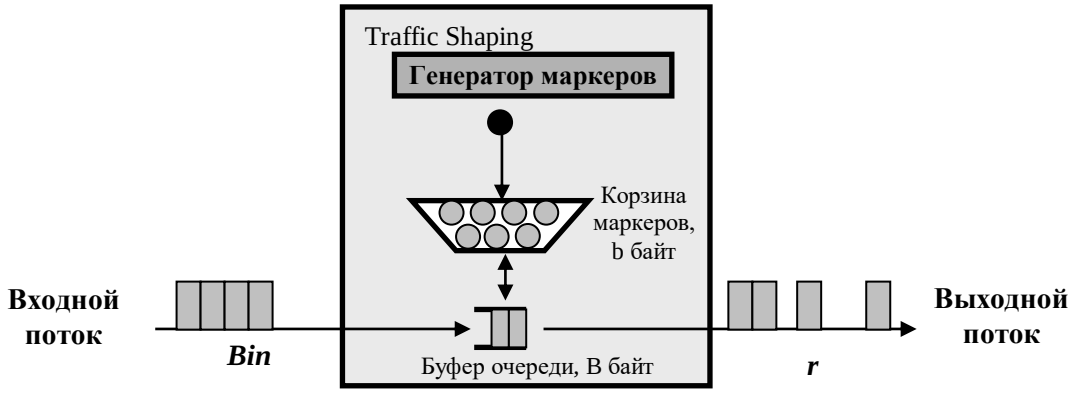

Рис. 2.26. Схема корзины маркеров

При *B<b* задержки в передаче пакетов отсутствуют, но в корзине всегда остаются неиспользованные маркеры. Из-за небольшого объема буфера очереди существует опасность потери пакетов во время пиковой нагрузки.

При *B=b* задержки в передаче пакетов отсутствуют, всегда происходит опустошение корзины маркеров. Для избегания потерь пакетов интенсивность поступления пакетов на входе должна точно соответствовать согласованной скорости, что редко выполняется на практике.

При *B>b* происходит опустошение корзины маркеров. Во время пиковой входящей нагрузки последний пакет в буфере очереди задерживается примерно на *(B+b)/r* секунд. Буфер очереди рассчитывают так, чтобы избежать потери пакетов: *B=b(p-r)/r*, где *p=(B+b)/T0* – пиковая скорость входящего потока. При переполнении буфера очереди пакеты теряются.

Хосты заинтересованы передавать информацию с максимальной скоростью, которая, однако, может привести к ***перегрузке*** сети. Поскольку к началу сеанса связи хосты ничего не знают от пропускной способности сети, то они пытаются определить ее самостоятельно, используя алгоритм медленного старта (рис. 2.27). В начале сеанса связи в соответствии с алгоритмом медленного старта устанавливается размер окна, равным одному сегменту (win=1·MSS байт), а пороговое значение размера окна ssthresh=winr – принятому от получателя. При нормальном прохождении сегмента (получении подтверждения) передатчик каждый раз увеличивает размер окна вдвое, поэтому к моменту времени t=3 текущее значение окна передатчика будет равно win=8·MSS байт. Интенсивная передача пакетов приводит к увеличению очереди в проходных маршрутизаторах и снижению пропускной способности сети (пунктирная линия), что приводит к отбрасыванию девятого и последующих сегментов. Такой алгоритм отбрасывания последних пакетов при переполнении очереди получил название отбрасывания хвоста (tail drop).

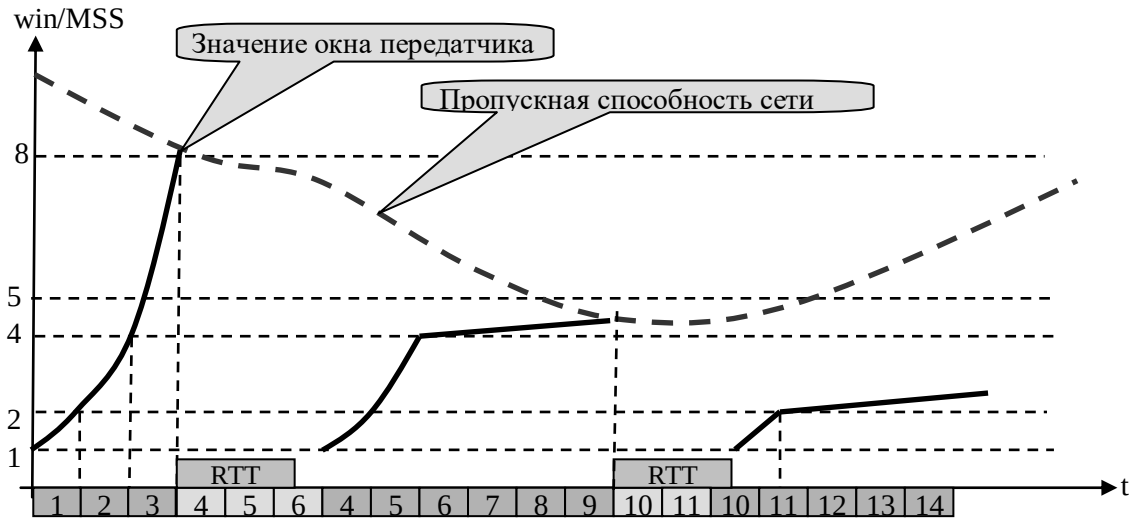

Рис. 2.27. Действие алгоритма медленного старта

Не получив в течение тайм-аута (Retransmit Timer Timeout, RTT) подтверждения о приеме 9-го сегмента, TCP-источник

сбрасывает значение окна до одного сегмента (win=1· MSS байт);

переустанавливает пороговое значение размера окна до величины, равной половине текущего значения (ssthresh=8· MSS байт/2=4·MSS байт), хотя значение окна, принятое от получателя winr может иметь гораздо большее значение;

возобновляет передачу неподтвержденных сегментов, следуя алгоритму медленного старта удваивая значение окна, пока оно не достигнет порогового значения (ssthresh=4· MSS байт). При достижении порогового значения размер окна начинает увеличиваться линейно с приращением MSS байт.

В случае повторной перегрузки описанная процедура повторяется, но с новым значением ssthresh.

Подобная процедура ограничения трафика устраняет перегрузку в момент фактического переполнения очереди и хорошо работает в сети с небольшим количеством пользователей. Реально же в сети обычно находится большое количество TCP-соединений. Получив уведомление о перегрузке (по истечению RTT), множество TCP-источников практически одновременно перезапустят алгоритм медленного старта – сбросят значение размера окна до одного сегмента и начнут экспоненциально его увеличивать, что неизбежно скоро приведет к новой перегрузке. Резкое периодическое снижение интенсивности трафика и его быстрый рост приводят к пилообразному изменению размера очереди, получившему название глобальной синхронизации (global synchronization).

Алгоритм отбрасывания хвоста приводит к резкому снижению пропускной способности сети. С целью устранения эффекта глобальной синхронизации применяют другой алгоритм управления длиной очереди – алгоритм произвольного раннего обнаружения перегрузки (Random Early Detection, RED). В соответствии с ним, вместо ожидания фактического переполнения очереди, алгоритм RED превентивно (заранее) начинает отбрасывать вновь поступающие пакеты с ненулевой вероятностью, зависящей от текущей длины очереди. Вероятностный характер отбрасывания пакетов приводит к отбрасыванию всего лишь нескольких произвольно выбранных пакетов, что является сигналом соответствующим TCP-источникам о необходимости перейти к перезапуску алгоритма медленного старта. Тем самым устраняется эффект глобальной синхронизации.

Отметим, что алгоритм RED не применим к протоколу UDP, который не чувствителен к потере пакетов. Имеется несколько модификаций алгоритма RED, связанных с селекцией пакетов по значению поля IP-приоритета, по принадлежности к определенному потоку (Weighted Random Early Detection, WRED) и т.д.

Механизм обслуживания очередей состоит из двух устройств – классификатора пакетов и планировщика очередей.

Классификатор пакетов (Packet Classifier) выполняет две функции: распознавание пакета и его маркировку (присвоение пакету определенного приоритета или класса). Он устанавливается в каждом пограничном маршрутизаторе. Функция распознавания зависит от пяти параметров – IP-адреса источника, IP-адреса назначения, порта источника, порта назначения, поля протокола IP-пакета (ICMP, TCP, UDP, …). В зависимости от принятого механизма обеспечения качества обслуживания пакет маркируется путем

установки поля IP-приоритета (биты 0-2 в поле TOS заголовка IP-пакета);

установки кода дифференцированной услуги (Differentiated Services Code Point, DSCP) взамен поля TOS в заголовке пакета;

внесения внутренней записи о принадлежности пакета к QoS-группе.

Планировщик очередей (Packet Scheduler) может использовать один из ниже перечисленных механизмов обслуживания очереди: взвешенного справедливого обслуживания потока (Weighted Fair Queuing, WFQ), взвешенного справедливого обслуживания классов (Class-Based Weighted Fair Queuing, CBWFQ), относительного приоритета (Priority Queue, PQ).

Планировщик очередей (Packet Scheduler) каждому пакету присваивает вес, значение которого используется для определения места в очереди на обслуживание.

При использовании механизма WFQ вес пакета (с последующим округлением до наиближайшего целого) определяется по формуле

$$W = \frac{4096}{IP + 1},$$

где W – вес пакета, IP – значение поля IP-приоритета (3 бита) в поле TOS (8 бит). Место в очереди на обслуживание вновь поступившего пакета вычисляется как произведение веса на длину пакета с учетом наличия в очереди ранее поступивших пакетов того же потока, т.е.

$$Q_x = \begin{cases} Q_{x-1} + W * LEN, & \textit{при наличии в очереди пакета того же потока}, \\ Q_t + W * LEN, & \textit{при отсутствии в очереди пакета того же потока}, \end{cases}$$

где Q_x – место ожидания вновь поступившего пакета, Q_{x-1} – место в очереди ранее поступившего пакета того же потока, LEN – длина пакета, Q_t – текущее место обслуживания. Для подсчета места ожидания используется циклический счетчик с максимальным значением несколько Гигабайт.

Пакеты одного потока помещается в индивидуальную выходную очередь. За один период прохода всех очередей считывается пакет с наименьшим местом ожидания. Упрощенно механизм WFQ можно представить как одну очередь, в которой место ожидания тем меньше, чем выше приоритет и короче длина пакета.

Рассмотрим прохождение очередей при поступлении пакетов разных приоритетов (рис. 2.28).

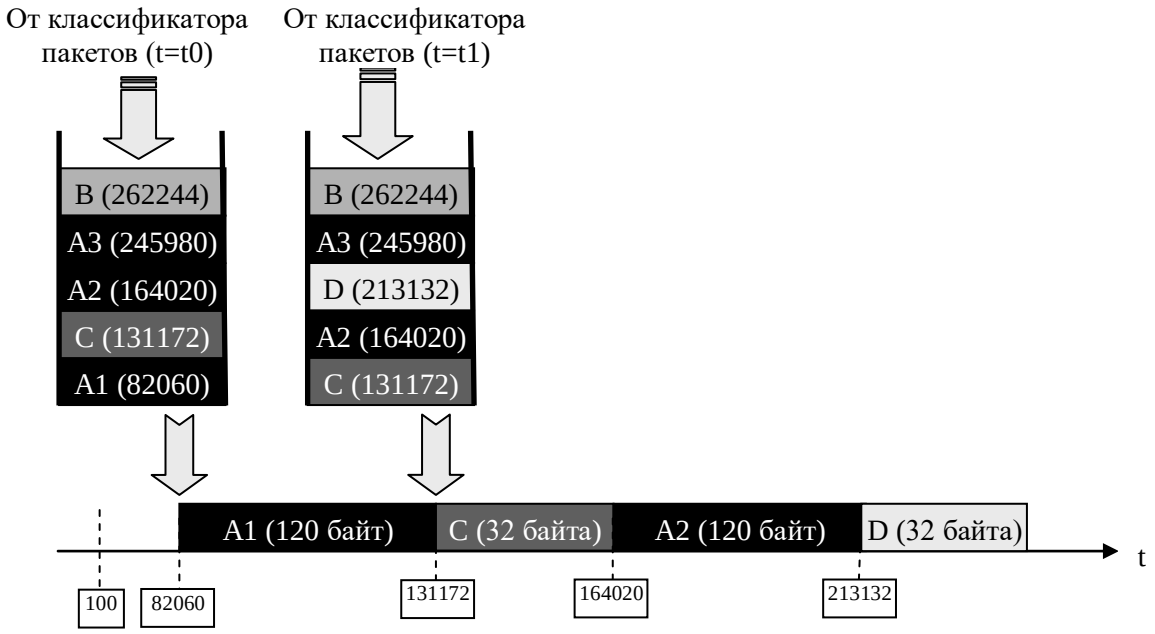

Рис. 2.28. Обслуживание пакетов планировщиком очередей

Предположим, что в момент времени t0 от классификатора пакетов в планировщик очередей поступают пакеты от трех потоков A, B и C. Текущее место обслуживания равно 100. Речевые пакеты потока A имеют приоритет 5 (вес=683) и длину 120 байт, пакет данных потока B имеет приоритет 0 (вес=4096) и дину 64 байта, пакет данных потока C имеет приоритет 0 (вес=4096) и дину 32 байта. Места в очереди планировщик распределит следующим образом: Q(A1)=100+(683·120)=82060, Q(C) = 100 + (4096·32) = 131172, Q(A2) = 82060 + (683·120) = 164020, Q(A3) = 164020 + (683·120) = 245980, Q(B) = 100 + (4096·64) = 262244. В момент времени t1, когда в обслуживании уже находится пакет A1, от классификатора поступает новый пакет D с приоритетом 0 (вес=4096) и диной 32 байта. Поскольку счетчик текущего места обслуживания к этому моменту уже установлен на значение 82060, то пакет D будет поставлен на Q=82060+(4096·32)=213132 место ожидания и будет обслужен раньше ранее поступивших пакетов A3 и B.

При использовании приоритетов на основании QoS-группы создается отдельная очередь, а вес пакета присваивается программой на основании какого-либо признака, не входящего в поле TOS или DSCP. Командой ***ip rsvp bandwidth*** *BW bwm* задается общая зарезервированная полоса *BW* и максимальное значение полосы *bwm*, которая может быть выделена отдельному приложению. Обычно $bwm \le BW \le 0.75 \cdot VEL$. Вес пакета определяется по формуле

$$W = \frac{4 * bwm}{bw},$$

где *bwm* – максимальное значение из всех полос, выделенных приложениям, *bw* – выделенная полоса передачи для конкретного приложения. Такое закрепление приоритетов используется в системе с интегральным обслуживанием.

При использовании механизма CBWFQ все пакеты, имеющие общий признак (IP-приоритет, поле DSCP и т.д.), относятся к одному классу. Каждому классу в явном виде выделяется определенная полоса, выраженная в абсолютных или относительных единицах измерения.

В системе с дифференцированным обслуживанием поле DSCP делится на две части, старшая часть (3 бита) определяет класс приоритета обслуживания, а младшая (3 бита) – приоритет обслуживания пакетов внутри класса (механизм обслуживания с абсолютным приоритетом PQ). Поле DSCP обладает обратной совместимостью с полем TOS, т.е. каждому значению класса приоритета поля DSCP соответствует значение TOS. Приоритет пакета определяется на основании кода DSCP, и он обслуживается с использованием механизма CBWFQ.

Пакеты одного класса одинаково маркируются *классификатором пакетов (Packet Classifier)*, а *планировщик очередей (Packet Scheduler)* помещает их в свою очередь. Более приоритетные пакеты помещаются в очередь с большей полосой передачи. В дальнейшем планировщик циклически обходит все очереди и считывает из каждой очереди число пакетов в соответствии с выделенной для нее полосой передачи.

Если *BWi* – абсолютная выделенная полоса для i-го класса, а *MTU* – максимальная длина пакета, то за один цикл из очереди i-го класса будет передано (считано) пакетов объемом $qi = \frac{BWi}{\min\{BWi\}} * MTU$ байт, где min{BWi} – минимальное значение из выделенных полос. Пусть, например, для классов A, B, C выделены полосы по 5, 60 и 35 Кбит/с, соответственно, а максимальная длина пакета на интерфейсе – 620 байт. Тогда за один цикл из очереди класса A будет считано (5/5)·620=620 байт, из очереди класса B – (60/5)·620=7440 байт, а из очереди класса C – (35/5)·620=4340 байт.

Если BWi% – относительное значение выделенной полосы для i-го класса, а *MTU* – максимальная длина пакета, то за один цикл из очереди i-го класса будет передано пакетов объемом $qi = \frac{BWi\%}{\min\{BWi\%\}} * MTU$ байт, где min{BWi%} – минимальное относительное значение из выделенных полос. Пусть, например, для классов A, B, C выделены полосы по 5%, 60% и 35% Кбит/с, соответственно, а максимальная длина пакета на интерфейсе – 620 байт. Тогда за один цикл из очереди класса A будет считано (5%/5%)·620=620 байт, из очереди класса B – (60%/5%)·620=7440 байт, а из очереди класса C – (35%/5%)·620=4340 байт.

Таким образом, чем выше приоритет, тем больше выделенная полоса передачи и, следовательно, тем быстрее пакет будет обслужен.

Механизм PQ реализует обслуживание с абсолютным приоритетом. Сетевой администратор определяет приоритетность обслуживания пакета путем внесения в базу данных классификатора записи вида "*IP-адресу=a.b.c.d* соответствует поле *DSCP=38*". Первые три бита поля DSCP определяют класс обслуживания (выделенную полосу передачи), последующие три бита – подкласс обслуживания. Внутри одного класса пакеты ставятся на различные места ожидания в соответствии с подклассом – high, medium, low. *Планировщик очередей (Packet Scheduler)* всегда начинает обслуживание очереди класса с подкласса high. Только после того, как все пакеты этого подкласса будут обслужены, начнется обслуживание подкласса medium. Соответственно, передача подкласса low окажется возможной только после того, как будут обслужены все более приоритетные подклассы. Вполне возможно, что в моменты перегрузки сети вся скорость передачи интерфейса будет занята высокоприоритетными пакетами, а передача пакетов подкласса low будет невозможна.

ВОПРОСЫ К РАЗДЕЛУ 2.4

1. Назовите обязательные и опциональные операции при формировании аудио и видео пакетов.

Ответ. Обязательные операции – выборка, квантование, кодирование, опциональная – сжатие.

2. Какой транспортный протокол применяется для переноса информации пользователя в реальном масштабе времени?

Ответ. UDP.

3. Поясните значение полос передачи второй строки табл.2.13 и первой строки табл.2.15.

Ответ. В первом случае каждые 20 мс передается пакет длиной 238 байт. Следовательно, за 1 секунду передается 238·1000/20 байт = 1190 байт = 95200 бит. Во втором случае за 30 мс передается пакет длиной 102 байта. Следовательно, за 1 секунду передается 102·1000/30 байт = 3400 байт = 27200 бит.

4. Возможен ли обмен информацией (например, разговор) между пользователями, использующими различные типы кодеков?

Ответ. Передающая и приемная части должны использовать один тип кодека.

5. Каким образом происходит устранение джиттера в IP-сети при приеме информации?

Ответ. На приеме используется сглаживающий буфер, при этом образуется дополнительная задержка, зависящая от емкости буфера.

6. Какова цель процедуры медленного старта?

Ответ. Выяснение хостом максимальной пропускной способности сети.

7. Предыдущий средний размер очереди – 300 байт. Текущий размер очереди 812 байт. Какой средний размер очереди?

Ответ. Средний размер очереди $B_{mid} = 300\cdot(1-2^{-9}) + 812\cdot 2^{-9} = 301$ байт.

8. На занятый выходной интерфейс поступает пакет с полем TOS=4 длиной 512 байт. Очередь на обслуживание отсутствует. На какое место ожидания пакет будет поставлен?

Ответ. Поступивший пакет будет иметь вес W=819 и будет поставлен на Qx = 0 + 819·512 = 419328 место ожидания.

9. На свободный выходной интерфейс поступает пакет с полем TOS=4 длиной 512 байт. Очередь на обслуживание отсутствует. На какое место ожидания пакет будет поставлен?

Ответ. Пакет будет обслужен немедленно.

2.5. Протоколы мультимедийных приложений

2.5.1. Классы сетевого качества обслуживания

В системе с интегральным обслуживанием (Integrated Services, Intserv) достижение требуемого качества обслуживания достигается предварительным закреплением ресурсов сети за устанавливаемым соединением. Процесс выделения ресурсов происходит при помощи ***протокола резервирования ресурсов (Resource Reservation Protocol, RSVP***). Состав механизма обеспечения качества обслуживания Intserv включает в себя (рис. 2.29): обработчик протокола RSVP, классификатор пакетов (Packet Classifier), планировщик очередей (Packet Scheduler), устройство политики (Policy Control), контроль доступа (Admission Control).

Во время поступления запроса на резервирование ресурсов (от своего или удаленного приложения) обработчик RSVP производит запрос базы данных *контроля доступа* о наличии ресурсов для резервирования (оставшейся неиспользованной полосы *BW*) и запрос базы данных *политики* о правомочности резервирования. Сведения об абоненте имеются либо в базе данных узла, либо в опциональном поле данных пакета *Policy data*, а содержимое данных политики вносит администратор сети. Если на оба запроса получены подтверждения, то в *классификаторе пакетов* производится соответствующая запись о закреплении ресурсов, иначе приложению возвращается сообщение об ошибке.

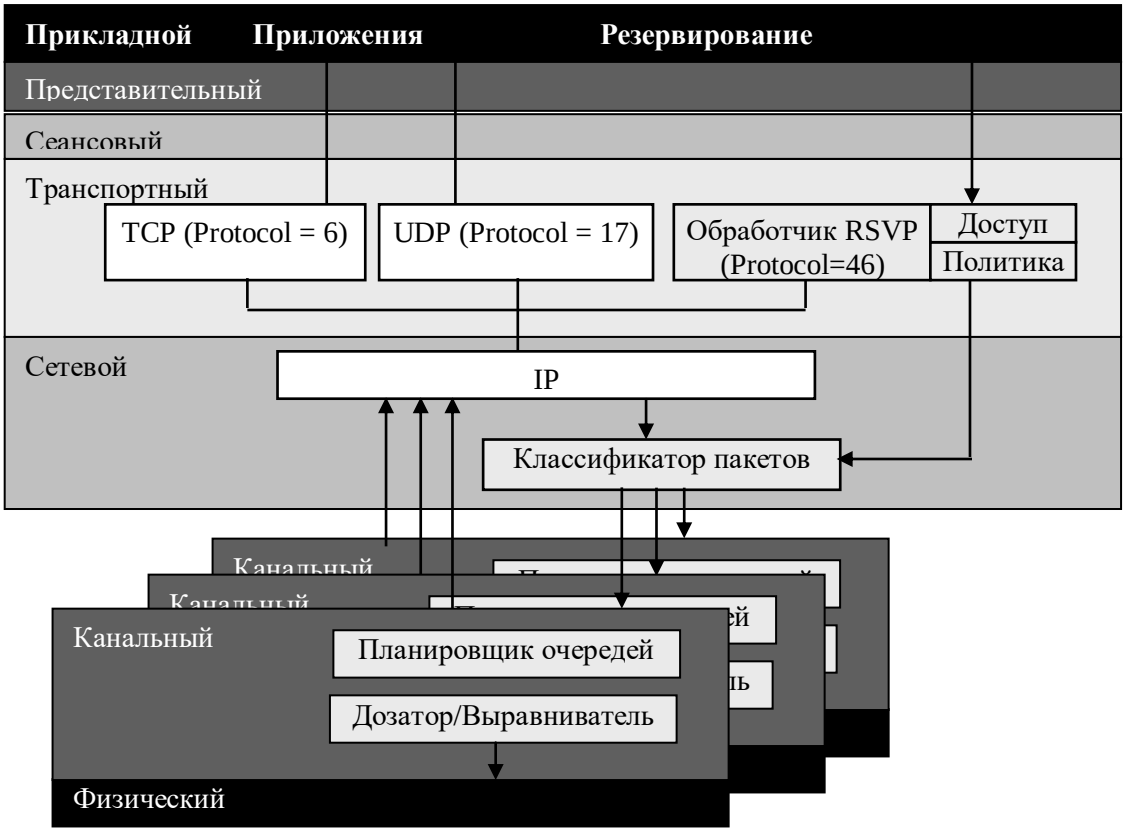

Рис. 2.29. Местоположение и состав механизма обеспечения качества обслуживания

В дальнейшем, при успешном резервировании, классификатор присваивает этому пакету (с известным ему IP-адресом, номером порта) определенный вес, а планировщик очередей обслуживает этот пакет в соответствии с заданным весом. После планировщика очередей устанавливается дозатор/выравниватель трафика, который сглаживает импульсивный характер трафика и следит за тем, чтобы суммарная скорость передачи всех пакетов не превышала бы фиксированную скорость передачи интерфейса.

Механизм QoS имеет следующие особенности (***RFC 2205*** [59]):

рассылка производится только в одном направлении;

для рассылки сообщений используются стандартные протоколы IPv4 или IPv6;

поддерживает индивидуальную и групповую рассылку своих сообщений, используя в последнем случае IGMP;

поддерживает прохождение сообщений через маршрутизаторы, не поддерживающих рассылку сообщений RSVP.

Местоположение и наименование полей RSVP изображено на рис. 2.30. Порядок передачи бит кадра: слева направо/снизу вверх, номер протокола – 46.

Поле описание объекта (Object)			
Время жизни (Send TTL)		Reserved	Длина сообщения (RSVP Length)
Версия (Version)	Флаги (Flags)	Тип сообщения (Message Type)	Контрольная сумма заголовка RSVP (Checksum)
3 уровень			
2 уровень			
1 уровень			

Рис. 2.30. Местоположение в стеке протоколов и формат RSVP

Назначение полей сообщения следующее.

Версия (Version) – определяет версию протокола, длина 4 бита (Version=1).

Флаги (Flags) – принимает значение 1.

Тип сообщения (Message Type) – принимает значение 1 для сообщения *Path*, 2 – для *Resv*, 3 – для *PathErr*, 4 – для *ResvErr*, 5 – для *PathTear*, 6 – для *ResvTear*, 7 – для *ResvConf*.

Контрольная сумма заголовка (RSVP Checksum) – используется для обнаружения ошибок при передаче сообщения, длина 16 бит.

Время жизни сообщения (Send TTL) – используется для обнаружения не RSVP-маршрутизаторов на пути следования сообщения путем сравнения значения TTL протокола RSVP со значением TTL протокола IP.

Длина сообщения (RSVP Length) – определяет длину заголовка и последующие поля описания объекта.

Поле описание объекта (Object) – содержимое полей зависит от типа сообщения.

Перед установлением соединения исходящий узел (на рис. 2.31 – хост 1) инициирует сообщение *Path*, которое в соответствии с IP-адресом доставляется корреспондирующему входящему узлу 2 через проходные сети N1 и N2 и проходные маршрутизаторы R1, R2, R3, R4, R5. Продвижение сообщения по сети осуществляют типовые IP-модули проходных маршрутизаторов. IP-модуль проходного RSVP-маршрутизатора, обнаружив тип протокола (Protocol=46), доставляет пакет обработчику RSVP, который 1)устанавливает программное состояние пути следования, запоминая IP-адрес предыдущего маршрутизатора, 2)вносит в сообщение необходимые дополнения и 3)возвращает сообщение IP-модулю для дальнейшей передачи. При обнаружении каких либо несоответствий в доступе/политике RSVP-маршрутизатор генерирует сообщение *PathErr* источнику рассылки. Не RSVP-маршрутизатор транслирует сообщение *Path* в соответствии с IP-адресом без обработки полей пакета.

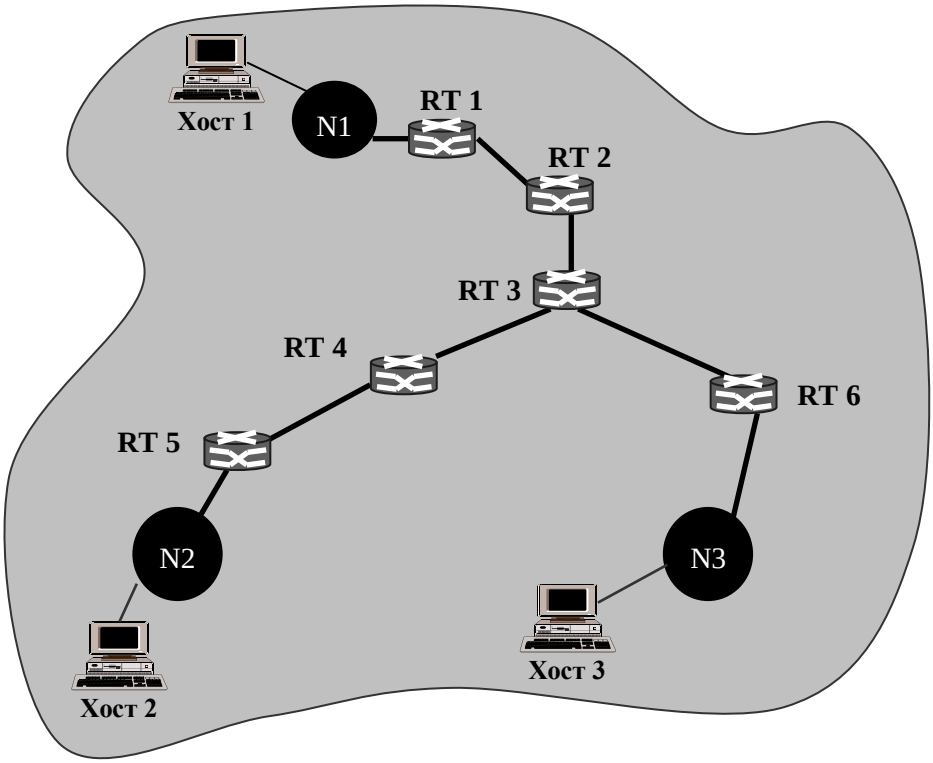

Рис. 2.31. Фрагмент сети с резервированием ресурсов

В ***сообщении Path*** в поле описания объекта помещается следующая обязательная информация (***RFC 2210*** [60]):

- SESSION. Содержит IP-адрес назначения, номер протокола, номер порта.
- RSVP_HOP. Содержит IP-адрес предыдущего (на рис. 2.31 – верхнего) маршрутизатора.
- TIME_VALUES. Содержит значение времени тайм-аута периодической посылки сообщения *Path* для сохранения программного состояния резервирования ресурсов (обычно 30 секунд).
- SENDER_TEMPLATE. Служит идентификатором потока для классификатора пакетов, содержит IP-адрес источника, номер протокола (TCP/UDP), номер порта.
- SENDER_TSPEC. Предназначено для планировщика очередей. Содержит описание параметров потока для реализации дозирования трафика по схеме “корзина маркеров” исходящего хоста, в частности: запрашиваемую полосу передачи – $r=bw$ (байт/с), размер корзины маркеров – b (байт), пиковую полосу передачи – p (байт/с), минимальный размер пакета – m (байт), максимальный размер пакета – M (байт).
- ADSPEC. Содержит базовую спецификацию параметров (Default Parameters), относящуюся ко всем классам обслуживания, и одну из дополнительных спецификаций запрашиваемого класса обслуживания: гарантированного обслуживания (Guaranteed Service) или контролируемой нагрузки (Controlled-Load). Служит для внесения уточнений параметров базовой спецификации каждым проходным RSVP-маршрутизатором.

Дополнительная спецификация включает в себя (***RFC 2215*** [62]):

1. Флаг наличия хотя бы одного не RSVP-маршрутизатора на всем пути. Функция формирования этого параметра – логическое ИЛИ. Установка этого флага производится при разнице полей TTL в протоколах IP и RSVP. Это означает, что хотя бы один маршрутизатор не поддерживает запрошенный класс обслуживания и поставлен в режим “с наибольшими усилиями”.
2. Число RSVP-маршрутизаторов на всем пути. Функция формирования этого параметра – арифметическое суммирование числа проходных RSVP-маршрутизаторов.
3. Фактически выделенная полоса передачи для всего пути – R (байт/с). За счет округления значения r до ближайшего целочисленного значения $R \geq r$ маршрутизаторы выделяют приложению обычно большую полосу передачи на определенный сервис, чем запрошенная полоса передачи – $r=bw$. Функция формирования этого параметра – минимальное значение.
4. Минимальная задержка на всем пути (мкс). Функция формирования этого параметра – арифметическое суммирование задержек, вычисленных каждым маршрутизатором при отсутствии очереди. Используется для определения минимального объема буфера для компенсации джиттера.
5. Максимальная длина пакета (байт). Функция формирования этого параметра – минимальное значение.

В ***классе гарантированного обслуживания (Guaranteed Service)*** каждый проходной маршрутизатор выделяет полосу передачи и объем буфера очереди не меньше запрошенных значений, т.е. $R \geq r$ и $B \geq (p-r)b/r$. Это позволяет обслуживать услугу с гарантированным качеством и без потерь. При недостаточности ресурсов запрос на резервирование отклоняется, а услуга обслуживается “с наибольшими усилиями”. Каждый сетевой элемент вычисляет минимальное значение C байт, необходимое для переприема (анализа адресной части пакета), а также значение аппаратной задержки D (микросекунд). При прохождении сообщения *Path* эти значения суммируются (***RFC 2212*** [61]) и вносятся в дополнительную спецификацию.

Максимальный объем буфера приемника для компенсации джиттера определяется как ($B + Ctot + Dtot \cdot R$). Максимальная задержка “из конца в конец” вычисляется как ($B/R + Ctot/R + Dtot$), где B – объем буфера очереди, R – фактически выделенная полоса передачи на всем пути, $Ctot$ – суммарное число байт переприема, $Dtot$ – суммарная аппаратная задержка.

В ***классе контролируемой нагрузки (Controlled-Load)*** каждый проходной маршрутизатор резервирует запрошенные ресурсы (r, b). В моменты перегрузки возможна частичная потеря пакетов. При отсутствии ресурсов услуга переводится в класс обслуживания “с наибольшими усилиями”. Оценка максимальной задержки и аппаратной задержки не производится. В дополнительную спецификацию вносится только флаг об отсутствии RSVP-маршрутизаторов на всем пути.

Входящий хост производит оценку объема необходимых ресурсов, которые отсылает источнику в ***сообщении Resv***. Сообщение *Resv* доставляется источнику по тому же пути следования, что и сообщение *Path*. IP-адрес назначения принимает значение предыдущего маршрутизатора, а IP-адрес источника – значение собственного адреса. При продвижении пакета *Resv* по пути следования (на рис. 2.31 – снизу вверх) каждый из проходящих RSVP-маршрутизаторов объединяет ресурсы, следующие в одном направлении, рассчитывает их достаточность и вносит запись в классификатор пакетов. Например, на рис. 2.31 маршрутизаторы R1, R2, R3 объединяют ресурсы при установлении соединения от хоста 1 к хостам 2 и 3. В результате сообщение ***Resv*** имеет следующие обязательные поля описания объекта:

- SESSION. Содержит IP-адрес назначения, номер протокола (TCP/UDP), номер порта.
- RSVP_HOP. Содержит IP-адрес предыдущего (на рис. 2.31 – нижнего) маршрутизатора.
- TIME_VALUES. Содержит значение времени тайм-аута периодической посылки сообщения *Resv* для сохранения программного состояния резервирования ресурсов (обычно 120 секунд).
- STYLE. Содержит стиль резервирования, т.к. объединение ресурсов не означает простого их суммирования, т.к. один из потоков может потребовать меньшую полосу передачи, но и меньшее значение задержки.
- FILTER_SPEC. Включает в себя данные, из принятого SENDER_TEMPLATE.
- FLOWSPEC. В зависимости от запрошенного класса содержит:

1. Для класса контролируемой нагрузки возвращает параметры потока TSPEC для реализации дозирования трафика по схеме “корзина маркеров”, в частности: запрошенную среднюю полосу передачи – r (байт/с), объем буфера корзины маркеров – b (байт), пиковую полосу передачи – p (байт/с), минимальный размер пакета – m (байт), максимальный размер пакета – M (байт). Пакеты, не соответствующие этому профилю, обслуживаются “с наибольшими усилиями” и могут быть утеряны.
2. Для класса гарантированного обслуживания в дополнение к перечисленным параметрам содержатся R (байт/с) – фактически выделенная полоса передачи на всем пути (R должно быть не менее r), S (мкс) – уменьшение суммарной задержки в обслуживании за счет смены параметра r на R. За счет выделения запрошенных ресурсов сетевая задержка минимальна, а потери пакетов исключены.

Сообщение ***ResvConf*** посылается в ответ на запрос подтверждения, содержащийся в сообщении ***Resv*** об успешном завершении процесса резервирования.

Сообщение ***ResvErr*** посылается любым промежуточным маршрутизатором при невозможности процесса резервирования.

Сообщение ***PathTear*** посылается исходящим хостом при окончании соединения или при истечении тайм-аута сохранения программного состояния. В поле описания объекта помещаются SESSION, RSVP_HOP, SENDER_DISCRIPTOR.

Сообщение ***ResvTear*** посылается при окончании соединения входящим хостом или при истечении тайм-аута программного состояния. В поле описания объекта помещаются SESSION, RSVP_HOP, FLOW_DISCRIPTOR.

В заключении следует отметить, что система с интегральным обслуживанием имеет достаточно сложный механизм резервирования ресурсов и применяется в отдельных зоновых сетях. В больших магистральных сетях (автономных системах) для обеспечения качества обслуживания обычно применяются системы с дифференцированным обслуживанием.

В системе с дифференцированным обслуживанием (Differentiated Services, Diffserv) согласно ***RFC 2474*** [70] повышение пропускной способности высокоприоритетных пакетов достигается присвоением им более высокого приоритета. Для этого используется поле IP протокола “Тип обслуживания” (Type of service, TOS), в котором 8 бит кода дифференцированной услуги (Differentiated Service Code Point, DSCP) используются следующим образом: 6 бит отводятся для кодирования приоритета, 2 – не используются и устанавливаются в “0”.

Провайдер услуг заключает с пользователем сети Интернет соглашение об уровне обслуживания (Service Level Agreement, SLA), на основании которого он составляет таблицу соответствия между уровнем обслуживания (максимальная полоса передачи, число пакетов в секунду) каждой услуги и полем приоритета DSCP. Эти данные администратор сети заносит в базу данных граничного маршрутизатора, к которому подключен пользователь. Такое же соглашение провайдеры различных смежных сетей устанавливают между собой. Граничный маршрутизатор отслеживает выполнение договора измерением поступающего трафика и маркирует каждый передаваемый пакет в соответствии с полем DSCP. Избыточный трафик может быть задержан, уничтожен или перемаркирован на другое значение поля приоритета.

Код дифференцированной услуги определяет цепочку решений о передвижении пакета в каждом промежуточном узле сети, так называемую PHB-политику (Per Hop Behavior). PHB-политика – это наблюдаемое извне поведение сетевого узла в отношении продвижения пакета с определенным значением кода дифференцированной услуги (DSCP). Все пакеты одного приложения имеют одно и то же значение DSCP (вес). В настоящее время коды DSCP и соответствующие им значения IP-приоритетов (TOS) для приложений определены следующим образом (см. табл. 2.18).

Табл. 2.18 – Коды DSCP

Класс приоритета	Значение DSCP	Значение TOS
Priority 7	111 000 (56)	7
Priority 6	110 000 (48)	6
Expedited Forwarding	**101 110 (46)**	5
Assured Forwarding Class 4 Class 4 High Class 4 Medium Class 4 Low	**100 000 (32)** 100 110 (38) 100 100 (36) 100 010 (34)	4
Assured Forwarding Class 3 Class 3 High Class 3 Medium Class 3 Low	**011 000 (24)** 011 110 (30) 011 100 (28) 011 010 (26)	3
Assured Forwarding Class 2 Class 2 High Class 2 Medium Class 2 Low	**010 000 (16)** 010 110 (22) 010 100 (20) 010 010 (18)	2
Assured Forwarding Class 1	**001 000 (8)**	1

Class 1 High Class 1 Medium Class 1 Low	001 110 (14) 001 100 (12) 001 010 (10)	
Best Effort	**000 000 (0)**	0

Стандартный код 000000, соответствующий стандартному обслуживанию (Best Effort), имеет наименьший приоритет (наибольший вес).

Коды обеспеченной передачи (Assured Forwarding). Они определяют четыре класса обслуживания, каждый из которых содержит три подкласса для реализации внутри одного класса обслуживания с абсолютным приоритетом.

Код немедленной передачи (Expedited Forwarding) имеет кодировку 101 110 и наивысший приоритет.

Механизм обеспечения качества обслуживания Diffserv включает в себя устройства:

- *классификатор пакетов (Packet Classifier),*
- *планировщик очередей (Packet Scheduler),*
- *устройство политики (Policy Control),*
- *контроль доступа (Admission Control).*

При заключении соглашения (SLA) с пользователем администратор сети вносит в базу данных *политики* и *контроля доступа* граничного маршрутизатора запись о закреплении ресурсов, т.е. запись о соответствии IP-адреса определенному значению DSCP. В дальнейшем, при поступлении данного типа пакета *классификатор пакетов* граничного маршрутизатора маркирует пакет в соответствии со значением DSCP, а *планировщики очередей* проходных маршрутизаторов обслуживают этот пакет в соответствии с механизмом CBWFQ. Пакеты, не принадлежащие SLA, обслуживаются в соответствии с механизмом “Best Effort”. После планировщика очередей устанавливается дозатор/выравниватель трафика, который сглаживает импульсивный характер трафика и следит за тем, чтобы суммарная скорость передачи всех пакетов не превышала бы фиксированную скорость передачи.

При взаимодействии автономных систем, имеющих различные классы сетевого качества обслуживания, требуется обеспечить сквозное качество обслуживания “из конца в конец”. Это достигается тем, что

система дифференцированного обслуживания транслирует протокол RSVP прозрачно;

классу гарантированного обслуживания (Intserv) ставится в соответствие класс немедленной передачи (Expedited Forwarding) Diffserv;

классу контролируемой нагрузки (Intserv) ставится в соответствие один из классов обеспеченной передачи (Assured Forwarding) Diffserv или стандартное обслуживание (Best Effort).

2.5.2. Технология коммутации по меткам

В традиционной IP-сети пакет, поступивший с некоторого интерфейса, поступает в модуль IP, который анализирует заголовок пакета и перемещает его на другой интерфейс в соответствии со своей таблицей маршрутов. Таблица маршрутов обычно заполняется автоматически с помощью протоколов маршрутизации (OSPF, RIP). Администратор сети имеет ограниченные возможности изменения таблицы маршрутов и не в состоянии оперативно повлиять на ситуацию в случае перегрузки какого-либо участка сети. Кроме того, просмотр всей таблицы для выбора оптимального маршрута часто приводит к значительной аппаратной задержке, снижающей производительность маршрутизатора. Поэтому желательно

- иметь возможность маршрутизировать пакеты, как на основе IP-адреса назначения, так и произвольным образом
- иметь упрощенную таблицу маршрутов установления соединений и перенести функции коммутации на более низкий уровень, освободив от этой работы IP-модуль.

Все эти возможности предоставляет технология ***многопротокольной коммутация по меткам (Multiprotocol Label Switching, MPLS)*** которая согласно ***RFC 3031*** [75] и ***RFC 3032*** [76] скорее является механизмом управления трафиком, чем механизмом обеспечения качества обслуживания. Технология MPLS используется маршрутизаторами внутри автономной сети, которую принято называть MPLS-сетью или MPLS-доменом. MPLS-маршрутизатор содержит две компоненты: распределитель меток и коммутатор меток (рис. 2.32).

Распределитель меток осуществляет распространение и привязку метки к интерфейсу, на который следует направлять пакеты, имеющих некое общее свойство (Forwarding Equivalence Class, FEC), например, префикс IP-сети. Основным протоколом распространения меток и их привязки к IP-адресам является протокол распространения меток (Label Distribution Protocol, LDP), использующий порт 646 протоколов UDP и TCP. Он ответственен за уникальность значений назначаемых меток и поддержание надежности функционирования. Последнее достигается периодическим обменом сообщениями *Hello* между сопряженными маршрутизаторами. *Коммутатор меток* осуществляет продвижение пакетов на основе замены меток, используя информационную базу меток (Label Information Base, LIB).

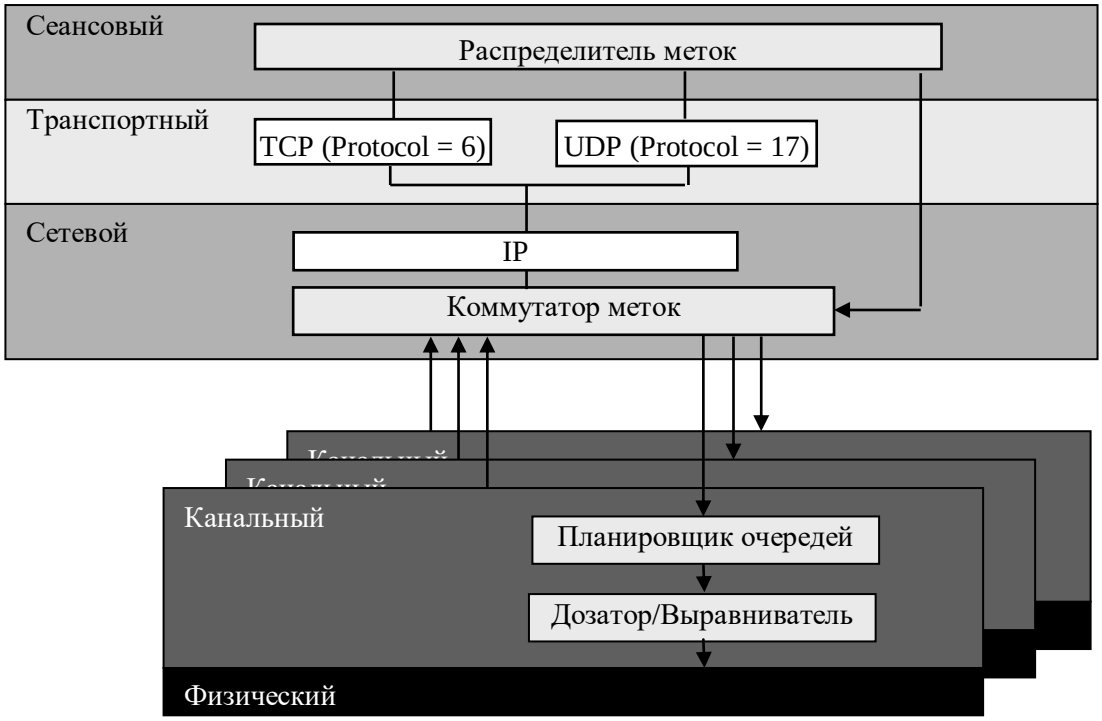

Рис. 2.32. Местоположение в стеке протоколов и состав механизма MPLS

Существует несколько способов привязки меток.

В режиме *нисходящего упорядоченного управления* инициатором распространения меток являются пограничные маршрутизаторы (Label Edge Router, LER). Распространение меток производится в направлении, противоположном движению потока. Каждый маршрутизатор имеет свой пул меток. Например, LER2 через интерфейс 2 имеет доступ к IP-сетям A и B (рис. 2.33). Через интерфейс 1 он распространяет эту информацию о сетях ближайшим соседям – внутреннему

маршрутизатору (Label Switching Router, LSR) LSR3. Признаком непосредственного подключения к LER2 указанных IP-сетей, является отсутствие в его сообщении метки (NULL).

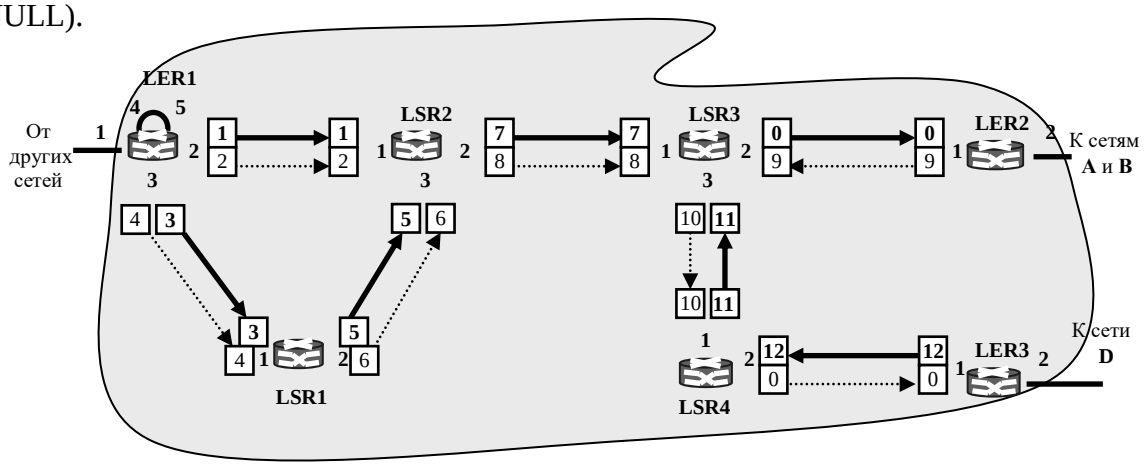

Рис. 2.33. Привязка меток в MPLS-сети

В свою очередь, LSR3 распространяет эту информацию на свои интерфейсы 1 и 3, указывая, что этим сетям соответствуют метки M7 и M11, соответственно. Так же поступают другие внутренние маршрутизаторы. В результате информация о привязке меток к IP-адресам достигает пограничных маршрутизаторов LER1 и LER3. Каждый маршрутизатор на основании принятой информации создает свою информационную базу меток. В каждой строке LIB содержатся соответствующие указания коммутатору меток. Значения меток для сетей A и B на этом этапе помечены на рис. 2.33 жирным шрифтом. Пусть, например, информационная база меток (LIB) для LER1 выглядит следующим образом (табл. 2.19):

Табл. 2.19 - LIB для LER1

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
1	A		PUSH	M1	2	3
1	B		PUSH	M1	2	3
1	A		PUSH	M3	3	11
1	B		PUSH	M3	3	11

которая указывает, что при поступлении пакетов на интерфейс 1, имеющих IP-адреса сетей A или B, следует создать (PUSH) метку M1 и отправить пакет на интерфейс 2. Здесь имеется в виду, что путь через интерфейс 2 имеет лучшую метрику “из конца в конец”, чем через интерфейс 3.

Стандартно значение сетевой метрики в таблице маршрутов IP модуля формируется на основе OSPF, а индивидуальные метрики интерфейса маршрутизатора в коммутаторе меток вносит администратор по следующей формуле: M=INT(100000/VEL), где VEL – скорость передачи по интерфейсу, измеренная в Кбит/с. Так, для интерфейсов со скоростями 100 Мбит/с и выше M=1, для канала 2048 Кбит/с M=48, для канала 64 Кбит/с M=1562 и т.д.

Пусть LIB для LSR2 выглядит следующим образом (табл. 2.20):

Табл. 2.20 - LIB для LSR2

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
1		M1	SWAP	M7	2	3
3		M5	SWAP	M7	2	3

которая указывает, что при поступлении пакетов на интерфейс 1, имеющих метку M1, следует заменить (SWAP) ее на метку M7 и отправить пакет на интерфейс 2. То же следует проделать с пакетом, имеющим метку M5 на интерфейсе 3.

Информационная база меток для LSR3 может выглядеть следующим образом (табл. 2.21):

Табл. 2.21 - LIB для LSR3

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
1		M7	POP		2	3
3		M11	POP		2	3

которая указывает, что при поступлении пакетов на интерфейс 1, имеющих метку M7, следует ее удалить (POP) и отправить пакет на интерфейс 2. При поступлении пакетов на интерфейс 3, имеющих метку M11, следует ее удалить (POP) и отправить пакет на интерфейс 2. Маршрутизатор LER2 принятые пакеты будет коммутировать на основе IP-адреса. Таким образом, от LER1 к сетям A, B образован однонаправленный основной коммутационный путь (Label Switched Path, LSP), проходящий через маршрутизаторы LER1-LSR2-LSR3-LER2 и резервный - LER1-LSR1-LSR2-LSR3-LER2.

В свою очередь, LER3 также распространяет информацию о сети D, подключенной к интерфейсу 2. Значения меток для сети D помечены на рис. 2.33 тонким шрифтом. Аналогично поступает и маршрутизатор LER1 для своих сетей (метки на рис. 2.33 для других сетей не показаны).

У администратора сети теперь появилась возможность гибкого перераспределения трафика (Traffic Engineering, TE). Например, в рассмотренном ранее случае предполагается, что путь LER1-LSR2-LSR3-LER2 обладает лучшей метрикой, чем путь LER1-LSR1-LSR2-LSR3-LER2. Но может случиться так, этот путь будет перегружен и администратор пожелает, чтобы в сеть A пакеты проходили по фиксированному пути LER1-LSR2-LSR3-LER2, а в сеть B – по другому фиксированному пути LER1-LSR1-LSR2-LSR3-LER2. В этом случае администратор может вручную изменить LIB для LER1 следующим образом (табл. 2.22):

Табл. 2.22 - Измененная LIB для LER1

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
1	A		PUSH	M1	2	3
1	B		PUSH	M3	3	11
1	A		PUSH	M3	3	11

которая указывает, что при поступлении пакетов на интерфейс 1, имеющих IP-адрес сети A, следует создать (PUSH) метку M1 и отправить пакет на интерфейс 2. При поступлении пакетов на интерфейс 1, имеющих IP-адрес сети B, следует создать (PUSH) метку M3 и отправить пакет на интерфейс 3.

Четырехбайтовая MPLS-метка имеет следующую структуру (табл. 2.23)

Табл. 2.23 – Структура MPLS-метки

Метка	CoS	S=1	TTL

в которой содержатся

- 20-битовая копия метки из LIB;
- 3-битовая поле класса обслуживания (Class of Service, COS), являющейся копией части поля TOS из заголовка IP-пакета. Предназначено для планировщика очередей;

- бит S, служит для создания стека MPLS-меток. При наличии нескольких меток только нижняя метка, расположенная ближе к сетевому (третьему) уровню имеет S=1, остальные метки имеют значение S=0;
- 8- битовая копия поля TTL из заголовка IP-пакета (уменьшенная на 1). Любой проходной LSR-маршрутизатор уменьшает значение TTL на 1, а предпоследний LSR-маршрутизатор возвращает поле TTL в заголовок IP-пакета.

Наличие стека меток позволяет администратору в *явном виде задавать любой маршрут от источника*. Например, для маршрута пакетов от LER1 к сети A и B по фиксированному пути LER1-LSR1-LSR2-LSR3-LER2 информационная база меток для LER1 будет иметь вид (табл. 2.24):

Табл. 2.24 – Создание стека меток LER1

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
1	A или B		PUSH	M7	4	0
5		M7	PUSH	M5	4	0
5		M5	PUSH	M3	3	11

которая при поступлении пакетов на интерфейс 1, имеющих IP-адрес сети A или B, через короткозамкнутые интерфейсы 4-5 создает стек меток M3, M5, M7 и отправляет пакет на интерфейс 3. В этом случае LSR1 будет иметь следующую LIB (табл. 2.25):

Табл. 2.25 – Работа со стеком LSR1

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
1		M3	POP		2	3

которая указывает, что при поступлении пакетов на интерфейс 1, имеющих метку M3, следует ее удалить (POP) и отправить пакет на интерфейс 2. На LSR2 поступит пакет с верхней меткой M5 и в соответствии со своей LIB (табл. 2.26)

Табл. 2.26 - Работа со стеком LSR2

ВХОД			ДЕЙСТВИЕ	ВЫХОД		
Интерфейс	IP-адрес	Метка		Метка	Интерфейс	Метрика
3		M5	POP		2	3

маршрутизатор LSR2 при поступлении пакета на интерфейс 3, имеющего метку M5, вытолкнет эту метку и направит пакет с меткой M7 на интерфейс 2. Маршрутизатор LSR3, как и раньше, просто вытолкнет последнюю метку M7 и направит пакет на интерфейс 2.

Каждый маршрутизатор должен самостоятельно определить наличие или отсутствие меток между 2-м и 3-м уровнями. Это особенно актуально для предпоследнего маршрутизатора, который должен удалить все метки вплоть до начала 3-го (сетевого) уровня. Для указания того, что пакет содержит метку (или стек меток) в протоколах канального уровня используются специальные поля. Так, обычный пакет использует типы протоколов LCP=0xc021, NCP=0x8021, Data=0x0021. Технология MPLS начинает работать после согласования параметров канального уровня (LCP), поэтому изменяются только поля NCP=0x8281 и Data (0x0281 для индивидуальной доставки или 0x0283 – для групповой). Замену указанных полей производят приграничные (LSR3 и LSR4) маршрутизаторы. Если MPLS используется для транспортирования пакетов через локальную сеть Ethernet, то в поле типа протокола подставляется значение 0x8847 (вместо 0x800) для индивидуальной доставки или 0x8848 – для групповой.

В режиме *восходящего управления по запросу* распространение меток производится в направлении движения потока. Инициатором распространения меток может быть любой маршрутизатор. Для этого он запрашивает у нижестоящего маршрутизатора метку и получает ее. В режиме *нисходящего упорядоченного управления по требованию* верхний маршрутизатор формирует запрос, который доставляется всем LER. В ответ каждый LER начинает раздавать метки, как и в режиме *нисходящего* упорядоченного управления.

Еще одним *способом распределения меток служит расширенный протокол RSVP* – RSVP-TE. Распределение меток инициируется исходящей стороной сообщением Path, которое продвигается либо на основе обычной маршрутизации, либо по заранее заданному маршруту полем Explicit Route Object (ERO). Ответное сообщение Resv содержит поле Label, содержащее входную метку нижележащего маршрутизатора.

2.5.3. Транспортный протокол реального времени

Для передачи мультимедийных приложений реального времени, в том числе речевых пакетов, используется ***транспортный протокол реального времени (Real time Transport Protocol, RTP)***. Местоположение в стеке протоколов и наименование полей RTP (***RFC 3550*** [81]) изображено на рис. 2.34. Порядок передачи бит кадра: слева направо/снизу вверх, номер порта – динамический.

Назначение полей сообщения следующее.

Версия (Version) – определяет версию протокола, длина 2 бита (Version=2).

Бит заполнения (Padding, P) – устанавливается в 1, если в поле полезной нагрузки (6-й уровень) внесены лишние дополнительные байты. Число внесенных байт содержится в последнем байте заполнения.

Бит расширения (Extension, X) – устанавливается в 1, если используется расширенный заголовок при экспериментальных расширениях RTP.

<table>
<tr><td colspan="7">Данные (Data)</td></tr>
<tr><td colspan="7">Лист идентификаторов нагрузки (Contributing Source (CSRC) identifiers)</td></tr>
<tr><td colspan="7">Идентификатор источника (Synchronization Source (SSRC) identifier)</td></tr>
<tr><td colspan="7">Временной штамп (Timestamp)</td></tr>
<tr><td>Версия (Ver)</td><td>P</td><td>X</td><td>Число источников (CC)</td><td>M</td><td>Тип нагрузки (PT)</td><td>Порядковый номер пакета (Sequence number)</td></tr>
<tr><td colspan="7">4 уровень (UDP)</td></tr>
<tr><td colspan="7">3 уровень</td></tr>
<tr><td colspan="7">2 уровень</td></tr>
<tr><td colspan="7">1 уровень</td></tr>
</table>

Рис. 2.34. Местоположение и формат RTP

Число источников (Contribution Sources Count, CC) – указывает на число используемых источников, участвующих в формировании поля полезной нагрузки, длина 4 бита. Устанавливается в 0 при соединении “точка-точка”.

Бит маркера (Marker, M) – устанавливается в 1 при наступлении значимого события, например, при окончании кадра.

Тип нагрузки (Payload Type, PT) – длина 7 бит. Указывает формат и способ кодирования/декодирования нагрузки: 0 – G.711m, 3 – GSM 13 Кбит/с, 4 – G.723, 8 – G.711a, 9 – G.722, 14 – MPEG Audio, 18 – G.729 и т.д.

Порядковый номер пакета (Sequence number) – начинается со случайного числа, увеличиваемого на 1 в каждом последующем пакете, длина 16 бит.

Временной штамп (Timestamp) – начинается со случайного числа, монотонно и линейно увеличиваемого во времени, длина 32 бита. Задает последовательность воспроизведения *Данных (Data).* Генерируется локальным счетчиком, входная частота которого зависит от типа нагрузки. Например, для аудио приложения входная частота равна частоте дискретизации, при времени формировании пакета 20 мс длиной в 160 байт каждое последующее значение штампа увеличивается на 160 единиц. Используется для вычисления емкости приемного буфера, компенсирующий разброс времени прихода пакетов относительно среднего значения.

Идентификатор источника (Synchronization Source identifier, SSRC) – случайное уникальное число, идентифицирующее источник нагрузки. При конференции принимает значение идентификатора устройства конференцсвязи.

Лист идентификаторов нагрузки (Contributing Source identifiers, CSRC) – опциональное поле списка идентификаторов источников, участвующих в формировании пакета, например, при конференции. Включает от 0 (при CC=0) до 15 полей длиной по 32 бита. Кажый источник имеет свое поле нагрузки. При конференции в поле SSRC помещается идентификатор устройства конференцсвязи, а в поле *CSRC* – идентификатор(ы) активного абонента.

Данные (Data) – содержит полезную нагрузку (речь), закодированную в соответствии с полем *Тип нагрузки.* Для каждого участника конференции создается свой поток с требуемым способом кодирования.

Протокол управления протоколом реального времени (Real time Control Protocol, RTCP) используется только совместно с RTP. Служит для рассылки источникам информации статистики качества прохождения пользовательской информации по протоколу RTP. Местоположение RTCP в стеке протоколов аналогично рис. 2.34, номер порта – динамический.

Имеется следующие типы RTCP-сообщений.

Отчет отправителя (Sender Report, SR) используется для передачи статистики от активных отправителей.

Отчет получателя (Receiver Report, RR) используется для передачи статистики от участников, которые не являются активными отправителями.

Описание источника (Source Description, SDES) содержит каноническое имя отправителя (Cname), символическое имя (Name) или адрес электронной почты (e-mail).

Окончание участия (BYE) используется в случае, если участник конференции покидает сессию.

Все типы сообщений передаются в одном составном пакете. Сообщения в составном пакете должны следовать в перечисленном порядке, т.е. первое сообщение всегда должно быть SR или RR, если даже не получено никаких пакетов или участник завершает соединение.

2.5.4. Протокол инициирования сеанса связи

Протокол инициирования сеанса связи (Session Initiation Protocol, SIP) согласно ***RFC 3261*** [78] является протоколом прикладного уровня поверх TCP или UDP (обычно UDP) и служит основой для построения сети мультимедийных соединений. Адресация в сети SIP происходит на основе универсального идентификатора ресурсов – Uniform Resource Identifier (SIP URI), который определяет символический адрес местонахождения коммутируемого сетевого элемента. Форма SIPS URI указывает на необходимость применения шифрования идентификатора на представительном уровне (Transport-layer security, TLS). Для описания параметров соединений (сессий) SIP-сервер использует текстовый (в коде ASCII) протокол описания сеансов связи (Session

Description Protocol, SDP), а в качестве системы адресации – форматы адресов, используемые в протоколах HTTP и DNS.

Составными частями сети являются:

- оконечные устройства (Terminals). Они обеспечивают двухстороннее соединение мультимедийной информации, представляют собой персональный компьютер, либо отдельное устройство, оснащенное мультимедийным приложением агента пользователя (User Agent, UA) SIP-протокола. Приложение UA состоит из клиентской части (User Agent Client, UAC) и серверной части (User Agent Server, UAS). Клиентская часть генерирует запросы, а серверная – формирует ответы на запросы;
- сервер SIP-сети (SIP Server). Является промежуточным сетевым элементом при взаимодействии различных агентов пользователей UA. Существуют три модификации сервера:

– сервер-посредник абонента (Proxy Server). Осуществляет поиск маршрута в сети от имени абонента. Имеет серверную часть (UAS) для приема запросов от абонента и клиентскую часть (UAC) для ретрансляции запроса следующему SIP-серверу;

– сервер переадресации (Redirect Server). Является сетевым элементом, принимая запрос от UA и возвращая ему список возможного местонахождения (SIP URI) следующего сетевого элемента. Имеет только серверную часть (UAS);

– сервер регистрации (Registrar Server). Хранит адреса постоянной регистрации (Address-of-Record, AOR) абонентов своего домена. Адрес постоянной регистрации содержит контактные адреса текущего (временного) местонахождение мобильного абонента. Обычно реализуется совместно с сервером-посредником или с сервером переадресации в виде единого устройства;

- мультимедийный шлюз (Media Gateway, MG). Обеспечивает сопряжение SIP-сети с другими типами сетей, например, с сетью коммутации каналов (Switched Circuit Network, SCN) и с другими шлюзами. Он производит преобразование протоколов разнородных сетей для установления и разрушения соединений, конвертирование мультимедийной информации пользователя в вид, пригодный для транспортировки по соответствующей сети.

Взаимодействие сетевых элементов происходит посредством обмена сообщениями, которые могут быть либо запросами, либо ответами. Предусматривается шесть видов запроса (Method):

REGISTER – для регистрации текущего местоположения абонента в сети,

INVITE, ACK, CANCEL – для установления (отказа от установления) соединения,

BYE – для разрушения установленного соединения,

OPTIONS – для запроса возможностей удаленного узла.

Ответы на запросы представляют собой 3-значный код, первая цифра определяет класс ответа, последующие цифры поясняют содержание. Предусмотрено шесть классов:

1xx – предварительный ответ (запрос принят и обрабатывается);

2xx – успешного завершения (запрос принят и обработан);

3xx – перенаправления вызова (запрос перенаправлен на другой сервер);

4xx – ошибки клиента (содержится ошибка в одном из полей запроса);

5xx – ошибки сервера (данный сервер не в состоянии обработать запрос);

6xx – глобальной ошибки (ни один из серверов не в состоянии обработать запрос).

Любое неопознанное содержание ответа UAC должен трактовать как x00 и обрабатывать соответствующим образом. Например, неопознанный ответ 211 должен трактоваться как 200 – успешное завершение.

Обязательными к распознаванию являются ответы 100 (trying, ждите) и 183 (session progress, запрос обрабатывается).

Входящее соединение к SIP-абоненту всегда устанавливается по известному адресу постоянной прописки (Address-of-Record, AOR) и проходит через домашний сервер-посредник (или сервер переадресации). При поступлении вызова домашний сервер-посредник (или сервер переадресации) обращается в домашний сервер регистрации для определения адреса текущего местонахождения пользователя – контактного адреса и, поучив его, устанавливает соединение на контактный адрес. Таким образом, сервер регистрации хранит таблицу соответствия адреса регистрации (Address-of-Record, AOR) пользователя в домене одному или нескольким адресам текущего местонахождения мобильного ресурса (контактные IP-адреса абонента).

Существует несколько способов составления таблиц соответствия. Например, они могут быть внесены администратором сети. Однако обычно используется механизм регистрации от пользователя. Для этого UA пользователя посылает запрос REGISTER серверу регистрации, который из запроса переписывает контактную информацию (***Contact:***).

Адрес сервера регистрации в UA вносится администратором сети, но в случае отсутствия такой записи UA в своем домене может определить адрес самостоятельно рассылкой запроса на регистрацию по групповому адресу 224.0.1.75. Приложение регистрации использует предопределенный порт с номером 5060 на обеих сторонах. Запрос на регистрацию содержит следующие поля.

- ***Via:*** – фиксирует путь прохождения запроса. Каждый проходной элемент добавляет в это поле транспортный протокол (UDP), свой SIP URI и параметр *branch*. Последний идентифицирует уникальный номер транзакции UA (задания для исполнения на удаленной стороне), на который ожидается ответ. Номер транзакции всегда должен начинаться со значения "z9hG4bK" и используется сервером для обнаружения “зацикливания” пакета в сети. Сервер регистрации копирует принятый IP-адрес источника и возвращает его параметром ***received*** в ответном сообщении.
- ***Max-Forwards:*** – определяет максимальное число проходных сетевых элементов. Начальное значение 70 каждый проходной уменьшает на единицу. При достижении значения 0 исходящему UA возвращается код ошибки 483.
- ***To:*** – адрес регистрации (Address-of-Record, AOR), имеет вид SIP URI или SIPS URI.
- ***From:*** – совпадает с полем ***To:,*** дополненное ярлыком (tag). Ярлык используется как часть идентификатора диалога (соединения) между передающей и принимающей сторонами.
- ***Call-ID:*** **–** уникальный идентификатор вызова, имеющий одно и то же случайное значение для всех последующих регистраций и установленных соединений. Рекомендуется использование формы "localid@host".
- ***CSec:*** – идентификатор порядкового номера регистрации для одного и того же идентификатора вызова ***Call-ID***. Содержит последовательный номер и тип запроса (method). В начале диалога принимает случайное значение, увеличиваемое на единицу в каждом последующем запросе.
- ***Contact:*** **–** содержит один или более контактных адресов. В последнем случае каждый адрес снабжается индивидуальным полем “*expires*” с соответствующим временем действия контактного адреса. Наиболее приоритетный адрес помечается более высоким параметром “q” – вероятностью нахождения пользователя по данному контактному адресу.
- ***Expires:*** **–** время (в секундах), по истечении которого регистрация недействительна для всех контактных адресок без индивидуального поля “*expires*”. Минимальное значение – 3600 (1 час). Для продления регистрации следует повторить процедуру, при отказе от регистрации следует установить значение “0”.

- ***Content-Length:*** – определяет длину сообщения, следующего за заголовком.

Для предотвращения несанкционированной регистрации, при которой входящее соединение могло бы устанавливаться на нелегального пользователя, применяется хорошо зарекомендовавшая себя схема аутентификации с цифровой подписью.

Пример регистрации пользователя с именем Bob приведен на рис. 2.35. Здесь сервер регистрации на сообщение REGISTER возвращает сообщение 401 (Unauthorized) со случайным числом (nonce), а пользователь во втором сообщении REGISTER высылает заверенный цифровой подписью дайджест и получает сообщение 200 OK.

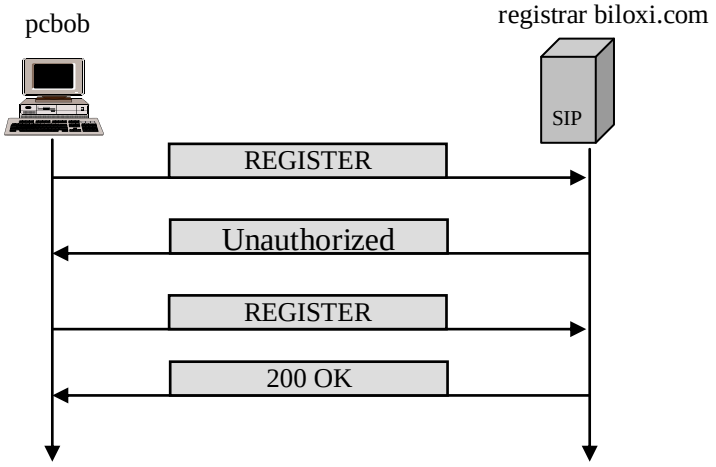

Рис. 2.35. Процесс успешной регистрации

Здесь в запросе (REGISTER) пользователь Bob сообщает домашнему серверу регистрации (доменное имя registrar.biloxi.com, порт 5060) адрес своего текущего местоположения (контактный IP-адрес=192.0.2.4, порт 5060).

Для надежного обмена в ***RFC 3261*** [78] предусмотрено подтверждение каждого запроса, но подтверждение доставки предварительных ответов не предусмотрено. Это негативно сказывается в сценариях длительного установления соединения, например, с абонентом, расположенным в телефонной сети общего пользования. Указанный недостаток устранен в ***RFC 3262*** [79], в котором для надежной доставки предварительных ответов предложено подтверждающее “из конца в конец” сообщение Provisional Response ACKnowledgement – PRACK. Инициирующая сторона указывает на возможности надежной доставки в поле Supported:100rel, а требование на надежную доставку – в поле Required:100rel. Если приемная сторона не поддерживает опции предварительных подтверждений, то она должна ответить сообщением 420.

Приемная сторона может запросить подтверждение приема предварительного ответа в запросе INVITE, за исключением подтверждения предварительного ответа 100 (TRYING), которое передается по звеньям. Причина введения предварительных подтверждений состоит в том, что обработка сообщения и получения на него подтверждения иногда занимает значительное время.

На рис. 2.36 приведен сценарий установления соединения между двумя узлами.

Первый узел представляет собой SIP-телефон с номером 123456789 (IP-адрес = 192.0.2.1), зарегистрированный в домене alabama.com. Второй узел представляет собой SIP-приложение на персональном компьютере с именем пользователя (адресом регистрации) bob@biloxi.com с контактным IP-адресом = 192.0.2.4, зарегистрированный в домене (сервере регистрации) biloxi.com. Соединение устанавливается с телефона с номером 123456789.

Рассмотрим содержимое сообщений рис. 2.36.

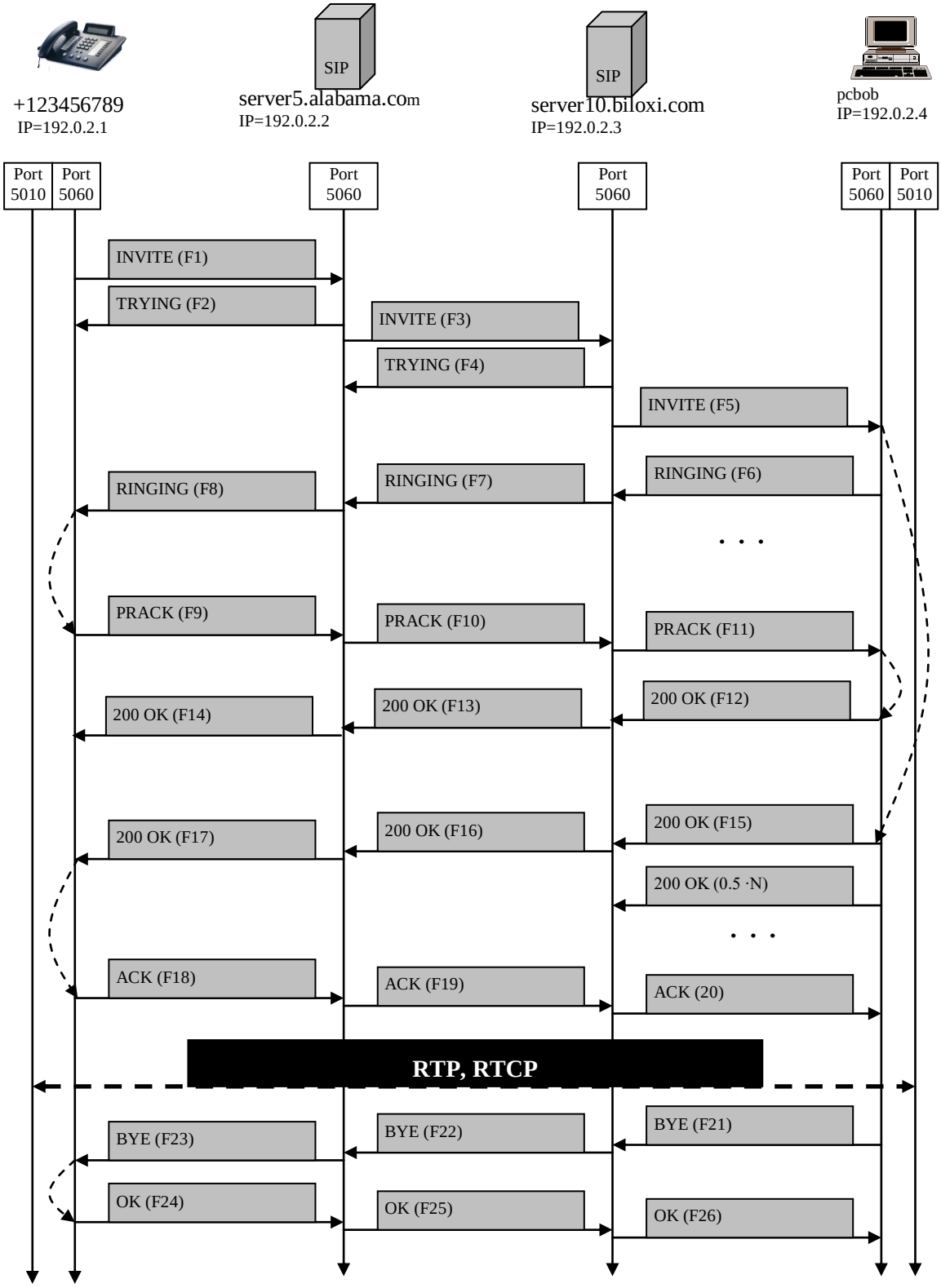

Рис. 2.36. Временная диаграмма установления SIP-соединения

F1 INVITE +123456789 → server5.alabama.com proxy

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
Supported:100rel
CSeq: 359 INVITE
Contact: <sip:+123456789@192.0.2.1:5060>
Content-Type: application/sdp
Content-Length: 156

V=0                          /версия SDP
o=MXSIP 0 4482 IN IP4 192.0.2.1          /тип оконечного устройства
s=SIP Call                   /вид вызова
c=192.0.2.1                  /IP-адрес
m=audio 5010 RTP/AVP 4 0 8          /вид связи, порт, условные номера кодеков
a=rtmap:4 G723/8000                 /параметры первого кодека
a=rtmap:0 PCMU/8000                 / параметры второго кодека
a=rtmap:8 PCMA/8000                 / параметры третьего кодека
```

INVITE – первое сообщение, посылаемое оконечным устройством, на адрес сервера-посредника домена alabama.com, которое будет устанавливать соединение. Клиентская часть UAC формирует

- номер транзакции (branch=z9hG4bKnashds8), на который ожидается ответ;
- ярлык (tag=1928301774), являющийся первой частью идентификатора диалога, идентифицирует номер участника диалога (при конференцсвязи участников может быть более двух);
- идентификатор вызова (Call-ID: a84b4c76e66710@alabama.com), являющийся третьей частью идентификатора диалога;
- порядковый номер запроса (CSec:359 INVITE);
- параметры описания сеанса связи (SDP).

Вторую часть идентификатора диалога должна прислать входящая сторона в поле To:.

F2 100 Trying server5.alabama.com.com proxy → +123456789

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 INVITE
Content-Length: 0
```

Сервер-посредник server5.alabama.com получил запрос, на который сразу ответить не может, т.к. не располагает информацией об IP-адресе вызываемого узла bob@biloxi.com. Поэтому он высылает ответ 100 (“Ждите”).

F3 INVITE server5.alabama.com.com proxy → server10.biloxi.com proxy

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
Max-Forwards: 69
To: Bob <sip:bob@biloxi.com>
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip:server5.alabama.com;lr>
```

Call-ID: a84b4c76e66710@alabama.com
Supported:100rel
CSeq: 359 INVITE
Contact: <sip:+123456789@ 192.0.2.1:5060>
Content-Type: application/sdp
Content-Length: 156
Содержимое SDP то же, что в сообщении F1

Сервер-посредник server5.alabama.com по имени bob@biloxi.com через DNS находит IP-адрес сервера-посредника в домене biloxi.com и делает к нему запрос с новым номером транзакции branch=z9hG4bK77ef4c2312983.1. Поле Record-Route: <sip:server5.alabama.com;lr> в заголовке пакета является опциональным. Оно предусматривает прохождение всех сигнальных сообщений сессии внутрь домена только через сервер server5.alabama.com. Решение о подстановке поля Record-Route принимается администратором домена, например, для тарификации установленных соединений, проходящих через домен, при использовании меж операторских взаиморасчетов. Параметр lr указывает, что заголовок маршрутизации соответствует ***RFC 3261*** [78].

F4 100 Trying server10.biloxi.com proxy → alabama.com proxy
SIP/2.0 100 Trying
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 INVITE
Content-Length: 0

F5 INVITE server10.biloxi.com proxy → Bob
INVITE sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
Max-Forwards: 68
To: Bob <sip:bob@biloxi.com>
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
Supported:100rel
CSeq: 359 INVITE
Contact: <sip:+123456789@192.0.2.1:5060>
Content-Type: application/sdp
Content-Length: 156
Содержимое SDP то же, что в сообщении F1

Сервер-посредник server10.biloxi.com, обращаясь к серверу регистрации пользователя registrar.biloxi.com (на рис. 2.36 не показано), получает контактный адрес и создает на него свой запрос узлу Bob с новым номером транзакции branch=z9hG4bK4b43c2ff8.1. Поле Record-Route: <sip: server10.biloxi.com;lr> предусматривает прохождение всех сообщений сессии из домена только через сервер server10.biloxi.com.

F6 180 Ringing Bob → server10.biloxi.com proxy
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1;received=192.0.2.3
Via: SIP/2.0/UDP server5.server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
Required:100rel
CSeq: 359 INVITE
RSec:125
Contact: <sip:bob@192.0.2.4:5060>
Content-Length: 0

Запрос на соединение получен узлом Bob. Абоненту Bob начинает поступать вызывной сигнал, генерируемый оконечным устройством. Узел формирует локальный ярлык tag=a6c85cf, который совместно с дистанционным ярлыком tag=1928301774 и идентификатором вызова Call-ID: a84b4c76e66710@alabama.com образуют идентификатор будущего диалога (сессии). Ответ 180 (“Вызов”) узел высылает серверу server10.biloxi.com. Поле Required:100rel указывает на необходимость подтверждения получения ответа 180 (“Вызов”).

F7 180 Ringing server10.biloxi.com proxy → alabama.com proxy

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
Required:100rel
CSeq: 359 INVITE
RSec:125
Contact: <sip:bob@192.0.2.4:5060>
Content-Length: 0

Поскольку ответ на транзакцию (задание) branch=z9hG4bK4b43c2ff8.1 получен, то сервер server10.biloxi.com изымает его из заголовка. Ответ 180 (“Вызов”) переправляется серверу server5.alabama.com.

F8 180 Ringing server5.alabama.com.com proxy → +123456789

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
Required:100rel
CSeq: 359 INVITE
RSec:125
Contact: <sip:bob@192.0.2.4:5060>
Content-Length: 0

Поскольку ответ на транзакцию (задание) branch=z9hG4bK77ef4c2312983.1 получен, то сервер server5.alabama.com изымает его из заголовка. Ответ 180 (“Вызов”) переправляется исходящему узлу с требованием подтвердить получение ответа.

F9 PRACK +123456789 → server5.alabama.com proxy

PRACK sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 360 PRACK
RAck: 125 359 INVITE
Content-Length: 0

В ответе 180 Ringing в поле Required:100rel имеется указание на подтверждение ответа. Поэтому исходящий хост посылает подтверждение получения ответа 180 Ringing.

F10 PRACK server5.alabama.com.com proxy → server10.biloxi.com proxy

PRACK sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 360 PRACK
RAck: 125 359 INVITE
Content-Length: 0

F11 PRACK server10.biloxi.com proxy → Bob

PRACK sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 360 PRACK
RAck: 125 359 INVITE
Content-Length: 0

Если бы предварительный ответ PRACK не был получен, то узел Bob через интервалы 0.5 с, 1 с, 2 с, … 16 секунд повторял бы отправку ответа 180 Ringing.

F12 200 OK Bob → server10.biloxi.com proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1;received=192.0.2.3
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 360 PRACK
Content-Length: 0

F13 200 OK server10.biloxi.com proxy → server5.alabama.com proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 360 PRACK
Content-Length: 0

F14 200 OK server5.alabama.com.com proxy → +123456789

SIP/2.0 200 OK

Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 360 PRACK
Content-Length: 0

F15 200 OK Bob → server10.biloxi.com proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1;received=192.0.2.3
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 INVITE
Contact: <sip:bob@192.0.2.4:5060>
Content-Type: application/sdp
Content-Length: 128
V=0
o=PC mw234 IN IP4 192.0.2.4
s=SIP Call
c=192.0.2.4
m=audio 5010 RTP/AVP 4 8
a=rtmap:4 G723/8000
a=rtmap:8 PCMA/8000

Абонент Bob ответил. Локальный ярлык tag=1928301774 совместно с дистанционным ярлыком tag=a6c85cf и идентификатором вызова Call-ID: a84b4c76e66710@alabama.com образуют идентификатор устанавливаемого диалога (сессии). В соответствии с протоколом входящий узел должен подтвердить установление сессии ответом OK. Он возвращает поля Record-Route и контактный адрес (Contact: <sip:bob@192.0.2.4:5060>). В своих параметрах описания сеанса связи (SDP) типы кодеков перечисляются в порядке их предпочтения.

F16 200 OK server10.biloxi.com proxy → server5.alabama.com proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 INVITE
Contact: <sip:bob@192.0.2.4:5060>
Content-Type: application/sdp
Content-Length: 128
Содержимое SDP то же, что в сообщении F15

F17 200 OK server5.alabama.com.com proxy → +123456789

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds8;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774

Record-Route: <sip: server10.biloxi.com;lr>
Record-Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 INVITE
Contact: <sip:bob@192.0.2.4:5060>
Content-Type: application/sdp
Content-Length: 128
Содержимое SDP то же, что в сообщении F15

F18 ACK +123456789 → server5.alabama.com proxy

ACK sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds9
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server5.alabama.com;lr>
Record-Route: <sip: server10.biloxi.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 ACK
Content-Length: 0

В соответствии с протоколом исходящий узел должен подтвердить установление сессии запросом ACK, на который не требуется подтверждения. К этому моменту исходящему узлу уже известен контактный адрес входящего узла Contact: <sip:bob@192.0.2.4:5060> и он мог бы выслать запрос ACK непосредственно входящему узлу. Однако принятое в ответе OK поле Record-Route предписывает высылать запросы через сервер server5.alabama.com. Узел формирует новый номер транзакции branch=z9hG4bKnashds9 и вместе с описанием маршрута следования (Record-Route) высылает запрос серверу server5.alabama.com. Локальный ярлык tag=1928301774 совместно с дистанционным ярлыком tag=a6c85cf и идентификатором вызова Call-ID: a84b4c76e66710@alabama.com образуют идентификатор диалога (сессии).

F19 ACK server5.alabama.com.com proxy → server10.biloxi.com proxy

ACK sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds9;received=192.0.2.1
Max-Forwards: 69
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Record-Route: <sip: server10.biloxi.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 ACK
Content-Length: 0

Сервер server5.alabama.com изымает из поля описания маршрута следования (Record-Route) собственный адрес и формирует новый номер транзакции branch=z9hG4bK77ef4c2312983.2, на который не ожидается ответ.

F20 ACK server10.biloxi.com proxy → Bob

ACK sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.2
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.2;received=192.0.2.2
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds9;received=192.0.2.1
Max-Forwards: 68
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: +123456789 <sip:+123456789@alabama.com;user=phone>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 359 ACK
Content-Length: 0

С получением этого сообщения диалог считается установленным. Если бы запрос ACK не был получен, то узел Bob через интервалы 0.5 с, 1 с, 2 с, … 16 секунд повторял бы отправку ответа OK. Узлы начинают обмен информацией по протоколу RTP/RTCP с использованием общего типа кодека G.723, стоящего первым в обоих списках SDP.

F21 Bye Bob → server10.biloxi.com proxy

BYE sip:+123456789@192.0.2.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: +123456789 <sip:+123456789@alabama.com>;tag=1928301774
Route: <sip: server10.biloxi.com;lr>
Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 231 BYE
Content-Length: 0

Пользователь Bob инициирует разъединение.

F22 Bye server10.biloxi.com proxy → server5.alabama.com proxy

BYE sip:+123456789@192.0.2.1:5060 SIP/2.0
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.3
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10;received=192.0.2.4
Max-Forwards: 69
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: +123456789 <sip:+123456789@alabama.com>;tag=1928301774
Route: <sip: server5.alabama.com;lr>
Call-ID: a84b4c76e66710@alabama.com
CSeq: 231 BYE
Content-Length: 0

F23 Bye server5.alabama.com.com proxy → +123456789

BYE sip:+123456789@192.0.2.1:5060 SIP/2.0
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.3
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.3;received=192.0.2.3
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10;received=192.0.2.4
Max-Forwards: 68
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: +123456789 <sip:+123456789@alabama.com>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 231 BYE
Content-Length: 0

F24 200 OK +123456789 → server5.alabama.com proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP server5.alabama.com;branch=z9hG4bK77ef4c2312983.3;received=192.0.2.2
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.3;received=192.0.2.3
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10;received=192.0.2.4
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: +123456789 <sip:+123456789@alabama.com>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 231 BYE
Content-Length: 0

F25 200 OK server5.alabama.com proxy → server10.biloxi.com proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.3;received=192.0.2.3
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10;received=192.0.2.4
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: +123456789 <sip:+123456789@alabama.com>;tag=1928301774

Call-ID: a84b4c76e66710@alabama.com
CSeq: 231 BYE
Content-Length: 0

F26 200 OK server10.biloxi.com proxy → Bob
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10;received=192.0.2.4
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: +123456789 <sip:+123456789@alabama.com>;tag=1928301774
Call-ID: a84b4c76e66710@alabama.com
CSeq: 231 BYE
Content-Length: 0

Во время установления соединения оконечные узлы могут изменять содержимое SDP при помощи сообщения UPDATE (***RFC 3311*** [80]).

Обычно адресация вызовов в сети SIP происходит на основе универсального идентификатора ресурсов – Uniform Resource Identifier (SIP URI) вида bob@biloxi.com, что позволяет определить текущее местоположение абонента через DNS. Для определения SIP URI оконечного устройства, имеющего телефонный номер в формате Рекомендации E.164, используется процедура ENUM (***RFC 2915*** [73]). Для этого в системе доменных имен имеется специальный домен “e164.arpa” (***RFC 3761*** [82]) со специальными записями – Naming Authority Pointer, NAPT. Домен имеет записи соответствия только для телефонных номеров в формате Рекомендации E.164. Например, для определения SIP URI по телефонному номеру +123456789 SIP-сервер осуществляет DNS-запрос (NAPTR RR query: 9.8.7.6.5.4.3.2.1.e164.arpa), на который получает ответ (например, bob@biloxi.com).

ВОПРОСЫ К РАЗДЕЛУ 2.5

1. В RSVP-маршрутизаторе зарезервирована общая полоса 1800 Кбит/с. Определите максимальное число соединений в системе с интегральным обслуживанием, если каждое приложение использует полосу 128 Кбит/с.

Ответ. Максимальное число соединений INT(1800/128)=14.

2. Чем отличается класс гарантированного обслуживания от класса контролируемой нагрузки в системе интегрального обслуживания?

Ответ. В классе гарантированного обслуживания производится 1)выделение полосы пропускания R и буфера очереди B с учетом пиковой нагрузки; 2)оценка максимальной задержки “из конца в конец”. В классе контролируемой нагрузки производится только выделение полосы пропускания r с учетом средней нагрузки, оценка максимальной задержки “из конца в конец” не производится.

3. Какое принципиальное отличие системы интегрального обслуживания от дифференцированного?

Ответ. Пакеты интегрального обслуживания получают практически всю скорость интерфейса, а пакеты дифференцированного обслуживания только ее часть.

4. Как обеспечивается сквозное качество обслуживания “из конца в конец” при взаимодействии автономных систем, имеющих различные классы сетевого качества обслуживания?

Ответ. Классу гарантированного обслуживания Intserv ставится в соответствие класс немедленной передачи (Expedited Forwarding) Diffserv. Классу контролируемой нагрузки (Intserv) ставится в соответствие один из классов обеспеченной передачи (Assured Forwarding) Diffserv.

5. Назовите основные преимущества применения MPLS.

Ответ. Уменьшение аппаратной задержки и возможность принудительной маршрутизации пакетов по заранее выбранному пути.

6. Поясните назначение протоколов RTP и RTCP.

Ответ. RTP переносит информацию пользователя и локальные временные отметки времени отправки пакета передающей стороной, позволяя управляющему устройству пользователя на приеме производить измерение джиттера. RTCP осуществляет извещение передатчика о качестве прохождения RTP-пакетов и позволяет измерить абсолютную задержку пакета “из конца в конец”.

7. Назовите состав сетевых элементов SIP-сети.

Ответ. Оконечные устройства пользователя; различные сервера SIP-сети для установления соединения; мультимедийные шлюзы для сопряжения с сетями связи, имеющими другие технологии передачи и коммутации.

8. Перечислите типы SIP-серверов и их основные функции.

Ответ. Сервер регистрации (Registrar Server) хранит адреса постоянной регистрации (Address-of-Record, AOR) и контактные адреса текущего (временного) местонахождение мобильного абонента. Сервер-посредник (Proxy Server) является промежуточным сетевым элементом при взаимодействии различных агентов пользователей и устанавливает соединение от имени пользователя. Сервер переадресации (Redirect Server) является сетевым элементом, принимая запрос на соединение от предыдущего сетевого элемента и возвращая ему список возможного местонахождения следующего сетевого элемента.

9. Какая информация содержится в поле Contact протокола SIP?

Ответ. Контактные адреса возможного нахождения абонента. Каждый контактный адрес имеет свой приоритет и срок действия.

10. Как организована адресация вызовов в сети SIP?

Ответ. На основе универсального идентификатора ресурсов – Uniform Resource Identifier (SIP URI).

11. Поясните назначение идентификатора вызова, номера транзакции и ярлыка.

Ответ. Идентификатор вызова (Call-ID) – уникальный номер соединения. Номер транзакции (branch) – номер текущего задания удаленной стороне для установления соединения. Ярлык (tag) – идентификатор пользователя, принадлежащего данному соединению (при конференцсвязи ярлыков может быть более двух).

2.6. Обеспечение безопасности сети связи

2.6.1. Общие сведения

В публичной сети (Интернет) возникает угроза несанкционированного доступа посторонних лиц к передаваемым данным, а также угроза перехвата передаваемого трафика. Для предотвращения этих угроз принимаются специальные меры, главными из которых являются:

- аутентификация – установление подлинности сообщения, в частности, автора. Эту задачу решают протоколы прикладного уровня;
- авторизация – контроль доступа легальных (аутентифицированных) пользователей к определенным ресурсам сети. Эта задача решается на прикладном уровне;
- шифрование – преобразование исходного текста, передаваемого по публичной сети, в “нечитаемый” для посторонних лиц текст. Эта задача, в зависимости от степени засекречивания, может решаться на разных уровнях. Если требуется защитить только тело сообщения прикладного уровня, например, электронной почты, то используется протокол S/MIME (Secure/MIME) (***RFC 2633*** [72]) или TLS (Transport Layer Security)

(***RFC 2246*** [63]). Если требуется защитить все поля передаваемого сообщения, то используется стандарт сетевого уровня IPsec (***RFC 2401*** [65], ***RFC 2406*** [66]).

Классическая модель криптосистемы приведена на рис. 2.37. В модели присутствуют три участника: два легальных пользователя и злоумышленник. Задача злоумышленника заключается в перехвате и чтении передаваемых сообщений. При этом предполагается, что злоумышленник имеет возможность подключения к каналу связи и ему доступны

1. Тексты передаваемых сообщений.
2. Алгоритм шифрования\дешифрования передаваемых сообщений.

Для предотвращения чтения перехваченных сообщений легальные пользователи используют шифрование. Шифрование состоит в том, что элементы исходного текста сообщения P различным образом “перемешиваются” с неизвестным злоумышленнику псевдослучайным текстом, называемым ключом K1 и в виде зашифрованного текста $C=E_{K1}(P)$ передаются в канал связи. Легальный пользователь, применяя к принятому сообщению C процедуру дешифрирования D ключом K2, получает исходный текст $D_{K2}(C)= D_{K2}(E_{K1}(P))=P$. В приведенных записях E и D – некие математические функции, а K1 необязательно равно K2. Значение ключа известно только легальным пользователям. Для повышения криптостойкости системы значения ключей периодически (например, раз в месяц) меняются. Существует несколько способов защиты информации.

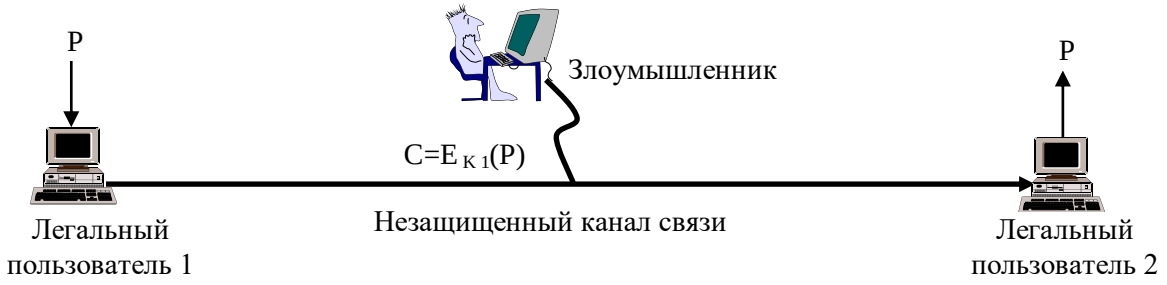

Рис. 2.37. Классическая модель криптосистемы

2.6.2. Методы защиты информации

Наиболее известным *симметричным алгоритмом шифрования с секретными ключами* является стандарт DES (**D**ata **E**ncryption **S**tandard). Каждая пара пользователей имеет один и тот же секретный ключ. Ключ представляет собой блок длиной 56 бит. Перед шифрованием любая форма представления исходных данных преобразуется в числовую. Данные шифруются поблочно, длина блока 64 бита. Блок делится пополам, на место левой половины блока помещается правая половина, а на место правой половины – сумма половин блока по модулю два. Затем по определенной схеме совершаются побитные замены некоторых элементов исходного текста на элементы ключа и их перемешивание. Для повышения крипто стойкости в настоящее время используется улучшенный стандарт шифрования AES (Advanced Encryption Standard), использующий симметричный ключ длиной от 128 до 256 бит.

В сети с N абонентами требуется N(N-1)/2 ключей, т.е. количество ключей возрастает пропорционально квадрату количества абонентов сети. Поэтому большую проблему представляет доставка секретных ключей такому большому количеству легальных пользователей. Для снятия этой проблемы пользователи используют несимметричные алгоритмы шифрования.

Несимметричный алгоритм шифрования с открытыми ключами основан на алгоритме Диффи-Хеллмана (W.Diffie, M.Hellman) и предполагает наличие у легального пользователя пары ключей таких, что текст, зашифрованный одним ключом, может

быть расшифрован только с помощью второго ключа. Один из ключей K1=e является открытым, его легальный пользователь раздает всем, от кого он хочет получать зашифрованную информацию. Другой ключ K2=d – закрытый, его пользователь хранит в защищенном месте. Наиболее известная реализация алгоритма, именуемая RSA, названа в честь его авторов R.**R**ivest, A.**S**hamir, L.**A**dleman и реализуется следующим образом.

1. Выбирается два простых числа, например, p = 7, q = 11.
2. Вычисляется произведение n = p·q = 7·11 = 77.
3. Вычисляется произведение m = (p-1)·(q-1) = 6·10 = 60.
4. Выбирается открытый ключ e = 13, как случайное число, не имеющее общих сомножителей с m = 60.
5. Вычисляется закрытый ключ d, как число обратное значению открытого ключа по модулю m, т.е. такое, что e·d mod m = 1, откуда d = 37.
6. Производится рассылка открытого ключа (n=77, e=13) легальным пользователям. Закрытый ключ (n=77, d=37) хранится в секрете.

Если легальный пользователь 1 собирается отправить пользователю 2 сообщение (например, символ A = 47 **не больше n**), то он шифрует его с использованием открытого ключа пользователя 2 как $B = A^e \bmod n = 47^{13} \bmod 77 = 5$. Пользователь 2 дешифрирует полученное значение B с помощью закрытого ключа как $B^d \bmod n = 5^{37} \bmod 77 = 47 = A$. Заметим, что дешифрировать сообщение B с помощью открытого ключа легальный пользователь 1 не может, т.к. $B^e \bmod n = 5^{13} \bmod 77 = 26 \neq A$.

Для повышения крипто стойкости длина ключа выбирается длиной не менее 1024 бит. В сети с N абонентами требуется по два ключа каждому абоненту: один открытый и один закрытый, всего 2N ключей. Шифрование и дешифрование по алгоритму RSA примерно в 1000 раз медленнее, чем по алгоритму DES.

Часто на практике используется *симметричный алгоритмом шифрования с открытыми ключами* по алгоритму Диффи-Хеллмана (Diffie-Hellman), который выглядит следующим образом.

Каждый пользователь сети имеет свой закрытый ключ Dj. Пусть, например, пользователь 1 имеет закрытый ключ D1=3. Инициатор соединения (пользователь 1) выбирает наугад два числа, например, g=5, n=17 и отсылает пользователю 2 значения g=5, n=17 и $y1= g^{D1} \bmod n = 125 \bmod 17 = 6$. Заметим, что значению y1=6 при известном g=5 и n=17 соответствует множество значений D1, а именно, 3, 19, 35, … Пользователь 2, имея закрытый ключ D2 = 7, проделывает аналогичную процедуру, высылая $y2 = g^{D2} \bmod n = 78125 \bmod 17 = 10$. Затем каждый пользователь в отдельности вычисляет общий ключ: пользователь 1 по формуле $K = (y2)^{D1} \bmod n = 10^3 \bmod 17 = 14$, а пользователь 2 – по формуле $K= (y1)^{D2} \bmod n = 6^7 \bmod 17 = 14$. В дальнейшем общий ключ используется как симметричный для двухстороннего обмена информацией. Для повышения криптостойкости 1)периодически меняют значения чисел g, n, Dj, 2)значения чисел g, n, Dj выбирают из большого диапазона случайных чисел, например, от 100 до 1024. В сети с N абонентами требуется всего N ключей.

Односторонний алгоритм шифрования применяется для проверки целостности данных. Существо преобразования состоит в том, что биты исходного текста P произвольной длины перемешиваются по определенному известному алгоритму с заранее известным бессмысленным текстом большой длины, поэтому даже короткий открытый текст преобразуется в текст гораздо большей длины. Затем вновь полученный текст сжимается по определенному известному алгоритму. Такое преобразование называется дайджест-функцией (digest function) или хэш-функцией, а результат – дайджестом H(P). Полученный дайджест намного короче исходного текста и по нему невозможно восстановить исходный текст. Дайджест является уникальной для данного

текста цифровой последовательностью, вероятность неопознанной подмены текста зависит от длины дайджеста A и вычисляется как 2^{-A}. Существуют два наиболее распространенных алгоритма вычисления дайджеста: MD5 (Message Digest 5) и SHA-1 (Secure Hash Algorithm 1). Первый из них формирует дайджест длиной 128 разрядов, второй – 160 разрядов.

Способ формирования дайджеста не является секретом, поэтому полученный дайджест H(P) шифруется с помощью секретного (или закрытого) ключа K, образуя цифровую подпись K(H(P)). Исходный открытый текст вместе с цифровой подписью передается в канал связи. Легальный получатель с использованием такого же секретного (или открытого ключа) дешифрирует цифровую подпись, получая исходный дайджест, а затем заново вычисляет дайджест из принятого открытого текста. Если вычисленный дайджест совпадает с исходным, принятым из канала связи, то принятый текст не был подвергнут никаким изменениям в процессе передачи, что дает получателю уверенность в целостности принятых данных.

2.6.3. Обеспечение безопасности на сетевом уровне

Обеспечение безопасности на сетевом уровне базируется на стандарте IPsec, в котором используется симметричный алгоритм шифрования с секретными ключами. Стандарт включает в себя три протокола: протокол AH (Authentication Header), обеспечивающего только целостность и аутентичность данных; протокол ESP (Encapsulating Security Payload), обеспечивающего целостность, аутентичность и шифрование данных; протокол IKE (Internet Key Exchange), обеспечивающий создание секретных ключей. IPsec может работать в двух режимах: транспортном и туннельном.

В транспортном режиме поле протокола IP-заголовка исходного пакета принимает значение 51, указывающее на наличие последующего заголовка протокола AH. Местоположение в стеке протоколов и наименование полей протокола AH в транспортном режиме приведено на рис. 2.38.

Следующий заголовок (Next header) – длина 8 бит, содержит код протокола вышележащего уровня (6 – TCP, 17 – UDP, …).

Длина поля нагрузки (Payload length) – длина 8 бит, содержит количество 32-х разрядных слов заголовка AH минус 2.

Указатель параметров безопасности (Security Parameters Index, SPI) – длина 32 бита, имеет ссылку на запись в базе данных получателя об использовании общего ключа и алгоритме формирования цифровой подписи (MD5 или SHA-1).

Порядковый номер (Sequence number) – длина 32 бита, содержит уникальный порядковый номер, даже если пакет посылается повторно. При достижении значения 2^{32} соединение закрывается и устанавливается заново.

Данные аутентификации (Authentication Data) – поле переменной длины, содержит цифровую подпись (зашифрованный дайджест), вычисленной с охватом всех полей исходного пакета за исключением поля *Время жизни,* которое изменяется при продвижении пакета по сети.

<table>
<tr><td colspan="3">Данные (Data)</td></tr>
<tr><td colspan="3">Данные аутентификации (Authentication Data)</td></tr>
<tr><td colspan="3">Порядковый номер (Sequence number)</td></tr>
<tr><td colspan="3">Указатель параметров безопасности (Security Parameters Index)</td></tr>
<tr><td>Следующий заголовок (Next header)</td><td>Длина поля нагрузки (Payload length)</td><td>Резерв (Reserved)</td></tr>
<tr><td colspan="3">Опции и заполнение (Options + Padding)</td></tr>
<tr><td colspan="3">Адрес назначения (Destination Address)</td></tr>
<tr><td colspan="3">Адрес источника (Source Address)</td></tr>
</table>

<table>
<tr><td colspan="2">Время жизни
(Time to Live)</td><td>Протокол
(Protocol = 51)</td><td colspan="2">Контрольная сумма
(Header Checksum)</td></tr>
<tr><td colspan="3">Идентификатор (Identification)</td><td>Флаги
(Flags)</td><td>Смещение фрагмента
(Fragment Offset)</td></tr>
<tr><td>Версия
(Version)</td><td>Длина
(IHL)</td><td>Тип услуги
(Type of Service)</td><td colspan="2">Общая длина
(Total Length)</td></tr>
<tr><td colspan="5">2 уровень</td></tr>
<tr><td colspan="5">1 уровень</td></tr>
</table>

Рис. 2.38. Местоположение в стеке протоколов и состав заголовка АН

другой внешний пакет с другим заголовком. Такое решение часто используется в виртуальных частных сетях (***Virtual Private Network, VPN***). Аутентифицируется весь пакет за исключением поля *Время жизни* внешнего пакета. В туннельном режиме местоположение протокола АН в пакете меняется, как показано на рис. 2.39.

Заголовок внешнего IP-пакета	Протокол АН	Заголовок исходного IP-пакета	Протоколы вышележащего уровня

Рис. 2.39. Местоположение протокола АН в туннельном режиме

В некоторых случаях, например, в виртуальных частных сетях (***Virtual Private Network, VPN***), требуется не только аутентификация, но и шифрование всего пакета, начиная от заголовка передаваемого пакета, из которого можно почерпнуть сведения о внутренней структуре сети, ее подсетях, полномочиях отдельных абонентов, активности деятельности предприятия, ее партнерах и т.д. Для этой цели используется протокол ESP, имеющий схожие с протоколом АН поля, но код поля протокола открытого IP-заголовка, указывающего на наличие последующего заголовка протокола ESP, принимает значение 50, а вычисленный дайджест (Данные аутентификации) помещается не в заголовок, а в конец пакета. Местоположение в пакете протокола ESP при транспортном и туннельном режимах показано на рис. 2.40. Поля, подлежащие шифрованию, выделены серым цветом.

Заголовок IP-пакета	Протокол ESP	Протоколы вышележащего уровня	Данные аутентификации

а)

Заголовок внешнего IP-пакета	Протокол ESP	Заголовок исходного IP-пакета	Протоколы вышележащего уровня	Данные аутентификации

б)

Рис. 2.40. Местоположение протокола ESP в а)транспортном и б)туннельном режимах

Существо туннелирования по протоколу ESP поясняет рис. 2.41. Зашифрованные поля помечены черным цветом.

Стандарт IPsec использует симметричный алгоритм шифрования с секретными ключами, поэтому еще до начала обмена информацией обе стороны должны удостовериться в легитимности сторон. Это осуществляется следующим образом.

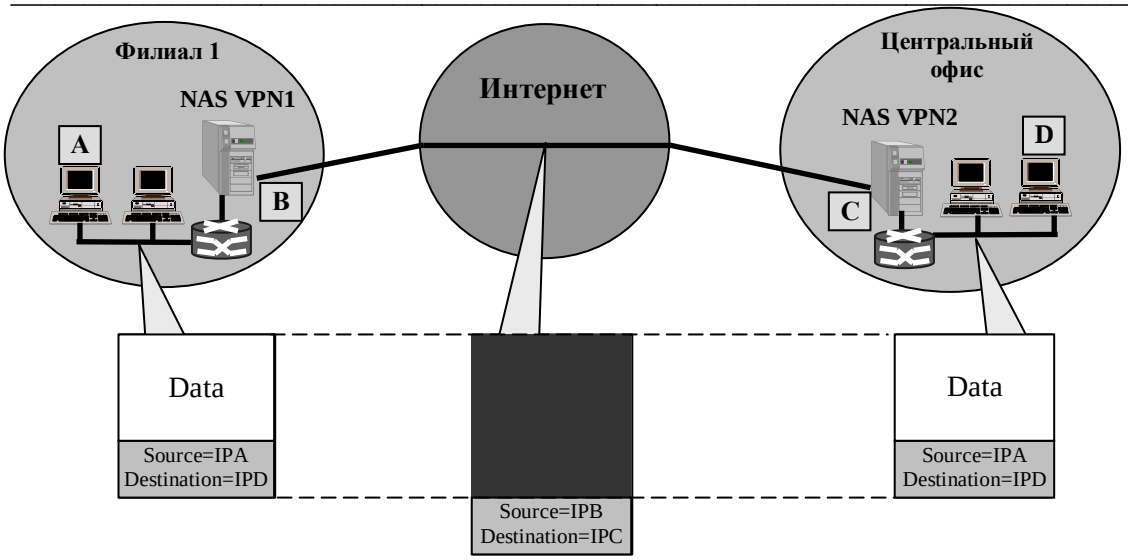

Рис. 2.41. Туннелирование пакетов

Вначале на обеих сторонах администраторы сети вручную создают идентичные базы данных, в которой каждому указателю параметров безопасности (Security Parameters Index, SPI) соответствует значение секретного ключа K и способ формирования дайджеста (MD5 или SHA-1). Процесс аутентификации и выбор параметров защищенного канала (безопасной ассоциации) показан на рис. 2.42.

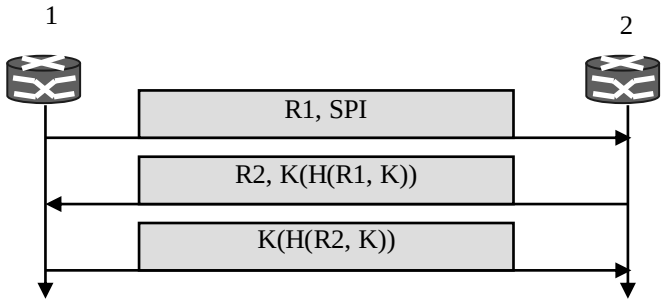

Рис. 2.42. Процесс обмена параметрами защищенного канала

В первом сообщении маршрутизатор 1 передает значение случайного числа R1 (например, временную отметку) и указатель параметров безопасности SPI. Маршрутизатор 2 по указателю параметров безопасности в своей базе данных находит значение секретного ключа K и способ формирования дайджеста (H). Затем он вычисляет дайджест из значений R1 и K, формирует цифровую подпись K(H(R1, K)) и вместе со своим случайным числом R2 возвращает их во втором сообщении маршрутизатору 1. Маршрутизатор 1, используя тот же ключ и способ формирования дайджеста, заново вычисляет цифровую подпись K(H(R1, K)) и сравнивает ее с принятой из канала связи. Их совпадение дает уверенность маршрутизатору 1 в легитимности маршрутизатора 2. Теперь маршрутизатор 1 должен доказать свою легитимность маршрутизатору 2, что он и делает с помощью третьего сообщения.

Изложенный метод используют сетевые администраторы для организации защищенной связи между смежными маршрутизаторами, а также для организации виртуальных частных сетей. Однако он неудобен пользователям, которым в большинстве случаев неизвестен секретный ключ вызывающего абонента. Поэтому пользователи используют другие методы, рассматриваемые в следующем разделе.

2.6.4. Обеспечение безопасности на представительном и прикладном уровнях

Аутентификация на представительном уровне обеспечивается путем обмена сертификатами при установлении соединения. Сертификат устанавливает соответствие принадлежности открытого ключа физическому или юридическому лицу. Организация, занимающаяся сертификацией открытых ключей, называется управлением сертификации (Certificate Authority, CA) и имеет иерархическую организационную структуру. Центральное управление сертификации сертифицирует региональные отделы, а региональные – местные отделения сертификации. Предполагается, что открытые ключи сертифицирующих органов для чтения сертификатов общедоступны и известны.

Способ описания сертификатов описан в ***Рекомендации ITU-T X.509*** [34]. Процедура получения сертификата начинается с того, что пользователь генерирует заранее открытый и закрытый ключи, а затем со своими паспортными данными и открытым ключом приходит в региональный отдел управления сертификации, которое выдает ему сертификат, подписанный своим закрытым ключом.

Сертификат содержит следующие поля (табл. 2.27).

Табл.2.27 – Содержание сертификата

Наименование поля	Значение поля
Version	Текущая версия – 3
serial Number	Серийный номер сертификата
Signature	Способ формирование дайджеста (MD5, SHA-1)
Issuer	Имя управления сертификации
Validity	Начало и конец срока действия
Subject	Сущность владельца ключа
subjectPublicKeyInfo	Значение открытого ключа
issuerUniqueIdentifier	Идентификатор, соответствующий issuer
subjectUniqueIdentifier	Идентификатор, соответствующий subject
Extensions	Расширения, в т.ч. доменные имена, IP-адреса, значения открытых ключей владельца сертификата и сертифицирующего органа

Секретность передаваемой информации обеспечивается симметричным алгоритмом шифрования, который обеспечивает высокую скорость шифрования. Значение общего ключа вычисляется каждой из сторон при каждом сеансе связи. Процесс установления защищенного соединения с использованием протокола TLS (Transport Layer Security) показан на рис. 2.43.

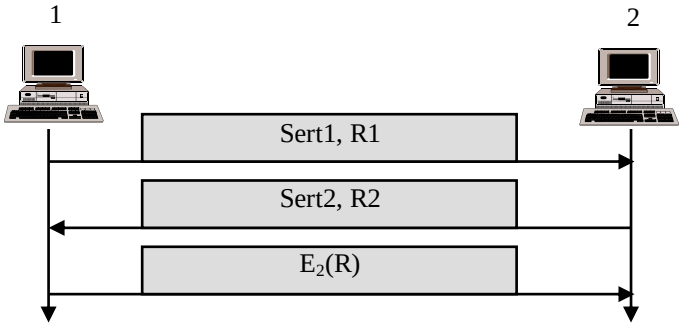

Рис. 2.43. Процесс обмена параметрами защищенного соединения

В первом сообщении пользователь 1 передает значение случайного числа R1 (nonce) и свой сертификат Sert1, содержащий открытый ключ E_1. То же самое во втором сообщении проделывает пользователь 2. Эти параметры передаются по сети в открытом виде. После получения второго сообщения пользователь 1 выбирает новое случайное число R (предварительный ключ), зашифровывает его открытым ключом E_2 , извлеченным из сертификата пользователя 2, и отправляет в сообщении 3.

Злоумышленник не в состоянии прочесть содержимое, т.к. он не является владельцем закрытого ключа D_2 пользователя 2. Значения R1, R2, R теперь известны на обеих сторонах и используются для генерации общего секретного ключа для данного сеанса связи.

Защита на прикладном уровне средствами протокола S/MIME (Secure/MIME) предполагает наличие у пользователя базы данных, в которой хранятся таблицы соответствия доменного имени абонента значению открытого ключа, извлеченного из легитимного сертификата управления сертификации.

Если у пользователя 1 имеется связка ключей {D1, E2}, а у пользователя 2 – {D2, E1}, где Ex – значение открытого ключа, а Dx – закрытого, то возможны следующие схемы шифрования/дешифрования сообщения P: E1(P) → D1(P), D1(P) → E1(P), E2(P) → D2(P), D2(P) → E2(P). Невозможны схемы E1(P) → E1(P), D1(P) → D1(P),

Схема формирования пользователем 1 цифровой подписи под открытым сообщением P показана на рис. 2.44. Здесь по исходному открытому тексту вычисляется дайджест H(P), который шифруется закрытым ключом пользователя D1, а затем кодируется в код base64. Таким образом, в канал связи передается открытый текст P, заверенный цифровой подписью пользователя 1 и сертификат.

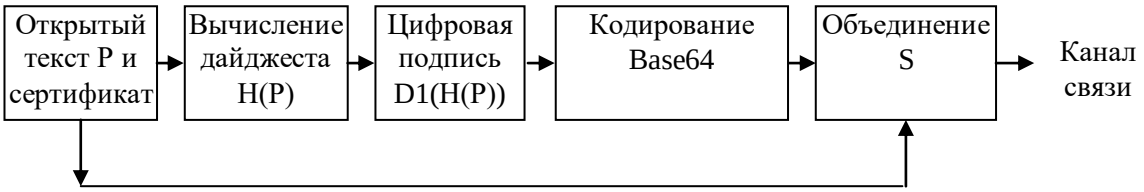

Рис. 2.44. Формирование цифровой подписи

При получении сообщений, сформированного в соответствии с рис. 2.44 , хост пользователя 2 по имени отправителя сообщения в своей базе данных находит открытый ключ пользователя 1 и, осуществляя обратные преобразования, проверяет подлинность открытого текста.

На рис. 2.45 приведена схема аутентификации и шифрования всего тела сообщения. По исходному тексту вычисляется дайджест H(P), который шифруется закрытым ключом пользователя D1. Затем открытый текст, сертификат и цифровая подпись пользователя 1 шифруются открытым ключом пользователя 2.

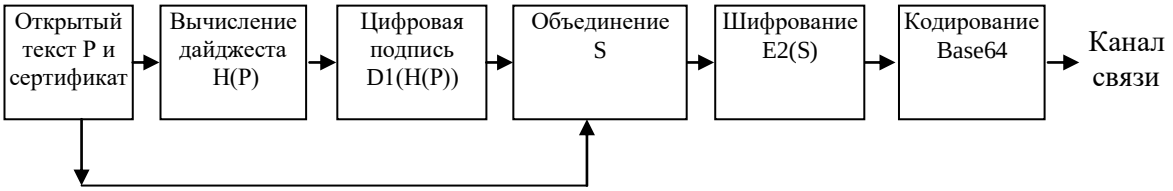

Рис. 2.45. Шифрование всего тела сообщения

При получении сообщений, сформированного в соответствии с рис. 2.45 , хост пользователя 2 по имени отправителя сообщения в своей базе данных находит открытый ключ пользователя 1 и, осуществляя обратные преобразования, получает исходный текст.

Может случиться так, что записи, соответствующей имени пользователя 1, в базе данных пользователя 2 может не оказаться. Тогда приложение должно уведомить пользователя и ждать от него инструкций. При получении согласия на прием, приложение вносит данные (имя пользователя и его открытый ключ), извлеченные из принятого сертификата, в базу данных и обрабатывает сообщение обычным образом.

2.6.5. Защита от DOS-атак

Для надежной доставки сообщений в IP-сетях используется TCP, который имеет следующие особенности, ограничивающий его применения для некоторых приложений:

1. Использование строгой последовательности доставки приводит к задержке последующих сообщений, если предыдущее было потеряно. Некоторые приложения не требуют строгой последовательности доставки. Это слишком серьезное ограничение для приложений, которые допускают как последовательную (частичное упорядочивание), так и непоследовательную доставку потоков.
2. Потоковая ориентация передаваемых сообщений часто неудобна приложениям, которые сами могут расставлять приоритеты доставки сообщений.
3. TCP имеет ограниченные возможности по множественной (multi homing) рассылке. Основанный на механизме TCP-сокетов API-интерфейс не поддерживает множественную адресацию. С конкретным TCP-соединением с другим хостом приложение может связать только один IP-адрес. Если интерфейс, назначенный этому IP-адресу, отключается, TCP-соединение прерывается и его необходимо устанавливать заново.
4. TCP чувствителен к перегрузкам сети.
5. TCP-хосты восприимчивы к атакам типа «отказ в обслуживании» (DoS — denial of service).

При нормальном соединении (рис. 2.46а) хост-отправитель сразу же производит резервирование ресурсов (памяти) для будущего соединения, а хост-получатель резервирует ресурсы после приема сообщения TCP SYN.

В отличие от нормального соединения для DoS-атак характерно огромное количество пакетов TCP SYN, поступающих от одного и того же хоста, но с разными “чужими” IP-адресами источника в течение короткого времени до истечения таймаута. Хост-получатель на каждый запрос резервирует память и отвечает сообщениями SYN ACK (рис. 2.46б), на которые не получает сообщения ACK, необходимые для завершения трехэтапной процедуры установки TCP-соединения. Ресурсы хоста, подвергнувшегося атаке, в течение некоторого времени (тайм-аута) не успевают освободиться и оказываются исчерпанными. Поэтому он оказывается не готов к обслуживанию легитимных запросов на установку TCP-соединения.

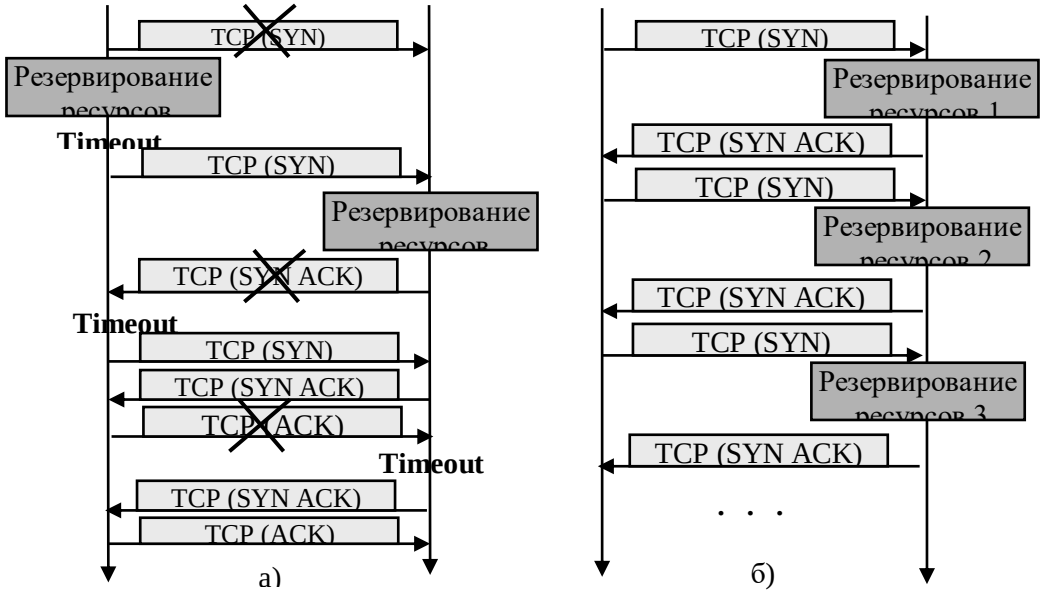

Рис. 2.46. Сценарии установления TCP-соединений
а) нормальное соединение со сбоями б)”отказ в обслуживании”

Передача телефонной сигнализации через IP-сеть является примером такого приложения, где указанные ограничения становятся существенны. Отличительные функции ***протокола управления передачей потока (Stream Control Transfer Protocol, SCTP)*** состоят в следующем.

1. Упорядоченная доставка фрагментов внутри потока. SCTP может обслуживать параллельно несколько потоков данных, в каждом потоке осуществляется упорядоченная доставка фрагментов. Это ускоряет доставку неповрежденных пакетов.

2. Фрагментация данных в соответствии с требуемым значением MTU. Механизм фрагментации аналогичен TCP.

3. Надежная доставка обеспечивается присвоением порядкового номера (Transmission Sequence Number, TSN) каждому фрагменту. Механизм защиты от перегрузки аналогичен TCP.

4. Возможность доставки в одном сообщении нескольких вложений (Chunk): данных и управления.

5. Контроль достоверности обеспечивается вычислением контрольной суммы.

6. Возможность обмена сообщениями по нескольким интерфейсам. Соединение TCP определяется как (IP_A+Port_A – IP_B+Port_B), где IP_A и Port_A это параметры одного конца соединения, а IP_B и Port_B – другого. Ассоциация SCTP определяется как (набор IP-адресов + Port_A) на одном хосте и (набор IP-адресов + Port_B) – на другом. При неработоспособности одного из интерфейсов (маршрутов) возможен обмен сообщениями по другому интерфейсу, который может использовать другой маршрут. Контроль работоспособности незагруженных интерфейсов обеспечивается их периодической проверкой (сообщения HEARTBEAT и HEARTBEAT ACK).

7. Защита от DoS-атак. В отличие от трехэтапной процедуры установки TCP-соединения SCTP предлагает четырехэтапную процедуру (рис. 2.47).

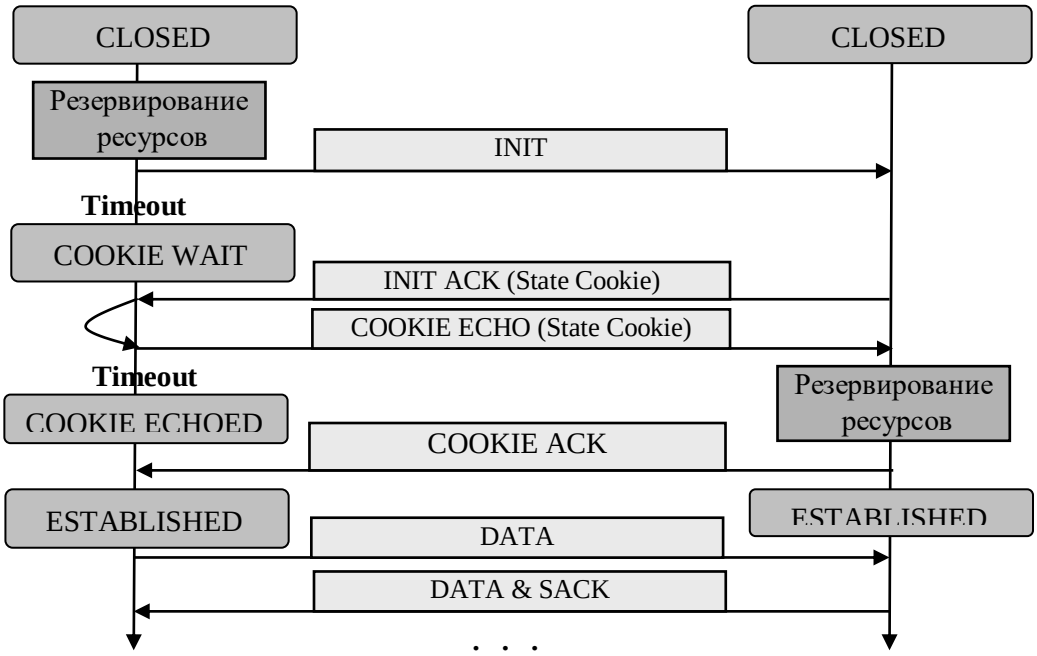

Рис. 2.47. Сценарий установления SCTP-соединения

В исходном состоянии CLOSED инициирующая сторона высылает сообщение INIT. Принимающая сторона, используя принятую информацию (IP-адреса источника и получателя, порта и т.д.) с помощью своего секретного ключа формирует дайджест (State Cookie) и отсылает его в сообщении INIT ACK обратно инициирующей стороне.

Никакие ресурсы не задействуются. Инициирующая сторона немедленно без изменений содержания возвращает принятое сообщение в виде COOKIE ECHO. Из этого сообщения (IP-адреса источника и получателя, порта и т.д.) приемная повторно вычисляет дайджест и сравнивает его с принятым в сообщении State Cookie. При совпадении дайджестов приемная сторона резервирует ресурсы, высылает согласие на соединение (COOKIE ACK) и начинает обмен данными (DATA) и подтверждать их прием (SACK).

Если инициирующая сторона подставит “чужой” адрес в сообщении INIT, то приемная сторона не получит сообщение COOKIE ECHO.

Если инициирующая сторона подставит “чужой” адрес в сообщении COOKIE ECHO, то приемная сторона это обнаружит по несовпадению дайджеста. В любом случае соединение не состоится, и ресурсы не задействуются.

Местоположение и наименование полей ***протокола управления передачей потока (Stream Control Transfer Protocol, SCTP)*** изображено на рис. 2.48. SCTP согласно ***RFC 2960*** [74] имеет номер протокола 132. Порядок передачи бит кадра: слева направо/снизу вверх.

Номер порта источника (Source Port Number) – определяет номер порта приложения, от которого поступил пакет, длина 16 бит.

Номер порта назначения (Destination Port Number) – определяет номер порта приложения, к которому следует доставить пакет, длина 16 бит.

Проверочный ярлык (Verification Tag) – определяет ассоциацию, к которой принадлежит пакет. Случайное число, длина 32 бита.

Контрольная сумма (Checksum) – используется для обнаружения ошибок при передаче пакета, длина 32 бита.

<table>
<tr><td colspan="2">. . .</td></tr>
<tr><td colspan="2">Chunk 1</td></tr>
<tr><td colspan="2">Checksum</td></tr>
<tr><td colspan="2">Verification Tag</td></tr>
<tr><td>Source Port Number</td><td>Destination Port Number</td></tr>
<tr><td colspan="2">3 уровень (Protocol=132)</td></tr>
<tr><td colspan="2">2 уровень</td></tr>
<tr><td colspan="2">1 уровень</td></tr>
</table>

Рис. 2.48. Местоположение и формат SCTP

За контрольной суммой следует один или несколько вложений (Chunk), каждый из которых имеет структуру рис. 2.49.

<table>
<tr><td>Chunk Type (Тип)</td><td>Chunk Flags (Флаги)</td><td>Chunk Length (Длина вложения)</td></tr>
<tr><td colspan="3">Chunk Value (Значение вложения)</td></tr>
</table>

Рис. 2.49. Формат вложенных полей протокола SCTP

Коды типов (Chunk Type) вложений приведены в табл. 2.28.

Табл. 2.28 – Коды и типы сообщений SCTP

Коды типов вложения	Наименование вложения
0	Полезная нагрузка (DATA)
1	Инициализация (INIT)
2	Подтверждение инициализации (INIT ACK)
3	Выборочное подтверждение (SACK)
4	Периодический запрос (HEARTBEAT)

5	Ответ на периодический запрос (HEARTBEAT ACK)
6	Отказ (ABORT)
7	Отключение (SHUTDOWN)
8	Подтверждение отключения (SHUTDOWN ACK)
9	Ошибка операции (ERROR)
10	Возврат дайджеста (COOKIE ECHO)
11	Подтверждение соединения (COOKIE ACK)
12,13	Резерв (Reserved)
14	Отключение завершено (SHUTDOWN COMPLETE)
15-255	Резерв (Reserved)

Число и значения флагов (Chunk Flags) зависят от типа вложения. Длина вложения определяется полем (Chunk Length), за которым следует его значение (Chunk Value).

Предполагается, что в будущем *протокол управления передачей потока (SCTP)* заменит TCP и UDP.

ВОПРОСЫ К РАЗДЕЛУ 2.6

1. Перечислите меры по обеспечению безопасности сети.

Ответ. Аутентификация – установление личности пользователя. Авторизация – доступ легальных (аутентифицированных) пользователей к определенным ресурсам сети. Шифрование – преобразование исходной информации в “нечитаемый” для посторонних лиц вид.

2. Какой протокол используется для аутентификации в публичной сети?

Ответ. CHAP.

3. Какой параметр используется для авторизации?

Ответ. Идентификатор (имя) пользователя.

4. Какой из методов шифрования является наиболее быстродействующим?

Ответ. Симметричный алгоритм шифрования с секретными ключами.

5. Какой метод шифрования требует наибольшего количества секретных ключей?

Ответ. Симметричный алгоритм шифрования с секретными ключами.

6. Какой метод шифрования текста используется при одностороннем алгоритме шифрования?

Ответ. Текст не шифруется, шифруется дайджест текста, который называется цифровой подписью.

7. Назовите стандартные алгоритмы формирования дайджеста.

Ответ. Алгоритм MD5 с длиной дайджеста 16 байт, алгоритм SHA-1 с длиной дайджеста 20 байт.

8. В каком методе обеспечивается большее сокрытие информации: транспортном или туннельном?

Ответ. В туннельном

9. Поясните механизм действия DOS-атак.

Ответ. Злоумышленник в течение короткого времени посылает на сервер большое количество запросов на соединение TCP SYN с разными “чужими” IP-адресами источника. Сервер на каждый запрос резервирует память, отвечает сообщениями SYN ACK и запускает таймер на ожидание подтверждения ACK, которые не приходят. На время истечения таймера ожидания подтверждения ресурсы памяти сервера оказываются исчерпанными.

10. Назовите основные преимущества протокола SCTP.

Ответ. Возможность обмена сообщениями по нескольким интерфейсам. Возможность доставки в одном сообщении нескольких вложений: данных и управления. Надежная защита от DoS-атак.